Lecture Notes in Computer Science 3762

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison
Lancaster University, UK

Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler
University of Surrey, Guildford, UK

Jon M. Kleinberg
Cornell University, Ithaca, NY, USA

Friedemann Mattern
ETH Zurich, Switzerland

John C. Mitchell
Stanford University, CA, USA

Moni Naor
Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz
University of Bern, Switzerland

C. Pandu Rangan
Indian Institute of Technology, Madras, India

Bernhard Steffen
University of Dortmund, Germany

Madhu Sudan
Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos
New York University, NY, USA

Doug Tygar
University of California, Berkeley, CA, USA

Moshe Y. Vardi
Rice University, Houston, TX, USA

Gerhard Weikum
Max-Planck Institute of Computer Science, Saarbruecken, Germany

Meersman Tari Herrero Méndez Cavedon
Martin Hinze Buchanan Pérez Robles
Humble Albani Dietz Panetto Scannapieco
Halpin Spyns Zaha Zimányi Stefanakis
Dillon Feng Jarrar Lehmann
de Moor Duval Aroyo (Eds.)

On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops

OTM Confederated International Workshops and Posters
AWeSOMe, CAMS, GADA, MIOS+INTEROP,
ORM, PhDS, SeBGIS, SWWS, and WOSE 2005
Agia Napa, Cyprus, October 31 - November 4, 2005
Proceedings

Springer

Volume Editors

Robert Meersman
Vrije Universiteit Brussel
STAR Lab, Pleinlaan 2, Bldg G/10, 1050 Brussels, Belgium
E-mail: meersman@vub.ac.be

Zahir Tari
RMIT University
School of Computer Science and Information Technology
City Campus, GPO Box 2476 V, Melbourne, Victoria 3001, Australia
E-mail: zahirt@cs.rmit.edu.au

Pilar Herrero
Universidad Politécnica de Madrid
Facultad de Informática,
Campus de Montegancedo S/N, 28660 Boadilla del Monte, Madrid, Spain
E-mail: pherrero@fi.upm.es

Library of Congress Control Number: 2005934895

CR Subject Classification (1998): H.2, H.3, H.4, C.2, H.5, I.2, D.2, K.4

ISSN 0302-9743
ISBN-10 3-540-29739-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-29739-0 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2005
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11575863 06/3142 5 4 3 2 1 0

OWeSOMe

Pilar Herrero
Gonzalo Méndez
Lawrence Cavedon
David Martin

CAMS

Annika Hinze
George Buchanan

GADA

Pilar Herrero
María S. Pérez
Víctor Robles
Jan Humble

MIOS + INTEROP

Antonia Albani
Jan L.G. Dietz
Herve Panetto
Monica Scannapieco

ORM

Terry Halpin
Robert Meersman

PhD Symposium

Antonia Albani
Peter Spyns
Johannes Maria Zaha

SeBGIS

Esteban Zimányi
Emmanuel Stefanakis

SWWS

Tharam Dillon
Ling Feng
Mustafa Jarrar
Jos Lehmann

WOSE

Peter Spyns
Aldo de Moor
Erik Duval
Lora Aroyo

OTM 2005 General Co-chairs' Message

The General Chairs of OnTheMove 2005, Agia Napa, Cyprus, are happy to observe that the conference series that was started in Irvine, California in 2002, and continued in Catania, Sicily, in 2003, and in the same location in Cyprus last year, clearly supports a scientific concept that attracts a representative selection of today's worldwide research in distributed, heterogeneous and autonomous yet meaningfully collaborative computing, of which the Internet and the WWW are its prime epitomes.

Indeed, as such large, complex and networked intelligent information systems become the focus and norm for computing, it is clear that there is an acute need to address and discuss in an integrated forum the implied software and system issues as well as methodological, theoretical and application issues. As we all know, email, the Internet, and even video conferences are not sufficient for effective and efficient scientific exchange. This is why the OnTheMove (OTM) Federated Conferences series has been created to cover the increasingly wide yet closely connected range of fundamental technologies such as data and Web Semantics, distributed objects, Web services, databases, information systems, workflow, cooperation, ubiquity, interoperability, and mobility. OTM aspires to be a primary scientific meeting place where all aspects for the development of internet- and intranet-based systems in organizations and for e-business are discussed in a scientifically motivated way. This fourth 2005 edition of the OTM Federated Conferences therefore again provides an opportunity for researchers and practitioners to understand and publish these developments within their individual as well as within their broader contexts.

The backbone of OTM is formed by the co-location of three related, complementary and successful main conference series: DOA (Distributed Objects and Applications, since 1999), covering the relevant infrastructure-enabling technologies; ODBASE (Ontologies, DataBases and Applications of SEmantics, since 2002), covering Web semantics, XML databases and ontologies; and CoopIS (Cooperative Information Systems, since 1993), covering the application of these technologies in an enterprise context through, e.g., workflow systems and knowledge management. Each of these three conferences encourages researchers to treat their respective topics within a framework that incorporates jointly (a) theory , (b) conceptual design and development, and (c) applications, in particular case studies and industrial solutions.

Following and expanding the model created in 2003, we again solicited and selected quality workshop proposals to complement the more "archival" nature of the main conferences with research results in a number of selected and more "avant garde" areas related to the general topic of distributed computing. For instance, the so-called Semantic Web has given rise to several novel research areas combining linguistics, information systems technology, and artificial intelligence,

such as the modeling of (legal) regulatory systems and the ubiquitous nature of their usage. We were glad to see that in 2005 under the inspired leadership of Dr. Pilar Herrero, several of earlier successful workshops re-emerged with a second or even third edition (notably WOSE, MIOS-INTEROP and GADA), and that 5 new workshops could be hosted and successfully organized by their respective proposers: AWeSOMe, SWWS, CAMS, ORM and SeBGIS. We know that as before, their audiences will mutually productively mingle with those of the main conferences, as is already visible from the overlap in authors!

A special mention for 2005 is again due for the second and enlarged edition of the highly successful Doctoral Symposium Workshop. Its 2005 Chairs, Dr. Antonia Albani, Dr. Peter Spyns, and Dr. Johannes Maria Zaha, three young and active post-doc researchers defined an original set-up and interactive formula to bring PhD students together: they call them to submit their research proposals for selection; the resulting submissions and their approaches are presented by the students in front of a wider audience at the conference, where they are then independently analyzed and discussed by a panel of senior professors (this year they were Domenico Beneventano, Jaime Delgado, Jan Dietz, and Werner Nutt). These successful students also get free access to "all" other parts of the OTM program, and only pay a minimal fee for the Doctoral Symposium itself (in fact their attendance is largely sponsored by the other participants!). The OTM organizers expect to further expand this model in future editions of the conferences and so draw an audience of young researchers into the OTM forum.

All three main conferences and the associated workshops share the distributed aspects of modern computing systems, and the resulting application-pull created by the Internet and the so-called Semantic Web. For DOA 2005, the primary emphasis stayed on the distributed object infrastructure; for ODBASE 2005, it became the knowledge bases and methods required for enabling the use of formal semantics, and for CoopIS 2005, the topic was the interaction of such technologies and methods with management issues, such as occur in networked organizations. These subject areas naturally overlap and many submissions in fact also treat an envisaged mutual impact among them. As for the earlier editions, the organizers wanted to stimulate this cross-pollination by a "shared" program of famous keynote speakers: this year we got no less than Erich Neuhold (Emeritus, Fraunhofer/IPSI), Stefano Ceri (Politecnico di Milano), Doug Schmidt (Vanderbilt University), and V.S. Subrahmanian (University of Maryland)! We also encouraged multiple event attendance by providing "all" authors, also those of workshop papers, with free access or discounts to one other conference or workshop of their choice.

We received a total of 360 submissions for the three main conferences and a whopping 268 (compared to the 170 in 2004!) in total for the workshops. Not only can we therefore again claim success in attracting an increasingly representative volume of scientific papers, but such a harvest of course allows the program committees to compose a higher quality cross-section of current research in the areas covered by OTM. In fact, in spite of the larger number of submissions, the Program Chairs of each of the three main conferences decided to accept only

approximately the same number of papers for presentation and publication as in 2003 and 2004 (i.e, average 1 paper out of 4 submitted, not counting posters). For the workshops, the acceptance rate varies but was much stricter than before, about 1 in 2-3, to almost 1 in 4 for GADA and MIOS. Also for this reason, we continue to separate the proceedings in two books with their own titles, with the main proceedings in two volumes, and we are grateful to Springer for their suggestions and collaboration in producing these books and CD-ROMs. The reviewing process by the respective program committees as usual was performed very professionally and each paper in the main conferences was reviewed by at least three referees, with email discussions in the case of strongly diverging evaluations. It may be worthwhile to emphasize that it is an explicit OTM policy that all conference program committees and chairs make their selections completely autonomously from the OTM organization itself. Continuing a costly but nice tradition, the OTM Federated Event organizers decided again to make all proceedings available as books and/or CD-ROMs to all participants of conferences and workshops, independently of one's registration.

The General Chairs are once more especially grateful to all the many people directly or indirectly involved in the set-up of these federated conferences and in doing so made this a success. Few people realize what a large number of individuals have to be involved, and what a huge amount of work, and sometimes risk, the organization of an event like OTM entails. Apart from the persons in their roles mentioned above, we therefore in particular wish to thank our 8 main conference PC Co-chairs (DOA 2005: Ozalp Babaoglu, Arno Jacobsen, Joe Loyall; ODBASE 2005: Michael Kifer, Stefano Spaccapietra; CoopIS 2005: Mohand-Said Hacid, John Mylopoulos, Barbara Pernici), and our 26 workshop PC Co-chairs (Antonia Albani, Lora Aroyo, George Buchanan, Lawrence Cavedon, Jan Dietz, Tharam Dillon, Erik Duval, Ling Feng, Aldo Gangemi, Annika Hinze, Mustafa Jarrar, Terry Halpin, Pilar Herrero, Jan Humble, David Martin, Gonzalo Médez, Aldo de Moor, Hervé Panetto, María S. Pérez, Víctor Robles, Monica Scannapieco, Peter Spyns, Emmanuel Stefanakis, Klaus Turowski, Esteban Zimányi). All, together with their many PCs, members did a superb and professional job in selecting the best papers from the large harvest of submissions. We also thank Laura Bright, our excellent Publicity Chair for the second year in a row, our Conference Secretariat staff and Technical Support Daniel Meersman and Jan Demey, and last but not least our hyperactive Publications Chair and loyal collaborator of many years, Kwong Yuen Lai.

The General Chairs gratefully acknowledge the logistic support and facilities they enjoy from their respective institutions, Vrije Universiteit Brussel (VUB) and RMIT University, Melbourne.

We do hope that the results of this federated scientific enterprise contribute to your research and your place in the scientific network... We look forward to seeing you again at next year's edition!

August 2005 Robert Meersman, Vrije Universiteit Brussel, Belgium
Zahir Tari, RMIT University, Australia
(General Co-chairs, OnTheMove 2005)

Organization Committee

The OTM (On The Move) 2005 Federated Workshops, involving nine workshops (i.e. AWeSOMe, Workshop on Agents, Web Services and Ontologies Merging; CAMS, Workshop on Context-Aware Mobile Systems; GADA, Workshop on Grid Computing and Its Application to Data Analysis; MIOS+INTEROP, Workshop on Inter-organizational Systems and Interoperability of Enterprise Software and Applications; ORM, Workshop on Object-Role Modeling; PhDS, PhD Symposium; SeBGIS, Workshop on Semantic-Based Geographical Information Systems; and SWWS, IFIP WG 2.12 & WG 12.4 Workshop on Web Semantics; WOSE, Workshop on Ontologies, Semantics and E-Learning) were proudly supported by RMIT University, Vrije Universiteit Brussel and Interop.

Executive Commitee

OTM 2005 General Co-chairs	Robert Meersman (Vrije Universiteit Brussel), Zahir Tari (RMIT University), and Pilar Herrero (Universidad Politécnica de Madrid)
AWeSOMe 2005 PC Co-chairs	Pilar Herrero (Universidad Politécnica de Madrid), Gonzalo Méndez (Universidad Complutense de Madrid), Lawrence Cavedon (Stanford University), and David Martin (SRI International)
CAMS 2005 PC Co-chairs	Annika Hinze (University of Waikato) and George Buchanan (University College London)
GADA 2005 PC Co-chairs	Pilar Herrero (Universidad Politécnica de Madrid), María S. Pérez (Universidad Politécnica de Madrid) Victor Robles (Universidad Politécnica de Madrid), and Jan Humble (University of Nottingham)
MIOS+INTEROP 2005 PC Co-chairs	Antonia Albani (University of Augsburg), Jan L.G. Dietz (Delft University of Technology), Hervé Panetto (University Henri Poincaré Nancy I), and Monica Scannapieco (University of Rome "La Sapienza")
ORM 2005 PC Co-chairs	Terry Halpin (Northface University) and Robert Meersman (Vrije Universiteit Brussel)
PhDS 2005 PC Co-chairs	Antonia Albani (University of Augsburg), Peter Spyns (Vrije Universiteit Brussel), and Johannes Maria Zaha (University of Augsburg)

SeBGIS 2005 PC Co-chairs	Esteban Zimányi (Université Libre de Bruxelles) and Emmanuel Stefanakis (Harokopio University of Athens)
SWWS 2005 PC Co-chairs	Tharam Dillon (University of Technology Sydney), Ling Feng (University of Twente), Mustafa Jarrar (STARLAB, Vrije Universiteit Brussel), Jos Lehmann (ISTC-CNR Rome), Aldo Gangemi (ISTC-CNR Rome), Joost Breuker (Leibniz Center for Law, The Netherlands)
WOSE 2005 PC Co-chairs	Peter Spyns (Vrije Universiteit Brussel), Aldo de Moor (Vrije Universiteit Brussel), Erik Duval (Katholieke Universiteit Leuven), and Lora Aroyo (Technische Universiteit Eindhoven)
Publication Co-chairs	Kwong Yuen Lai (RMIT University) and Peter Dimopoulos (RMIT University)
Organizing Chair	Skevos Evripidou (University of Cyprus)
Publicity Chair	Laura Bright (Oregon Graduate Institute)

AWeSOMe 2005 Program Committee

Richard Benjamins
Adam Cheyer
Ian Dickinson
Tim Finin
Hamada Ghenniwa
Jorge Gómez
Dominic Greenwood
Mike Huhns
Lewis Johnson
Margaret Lyell
Michael Maximilien
Juan Pavón
Terry Payne
José Peña
María Pérez
Debbie Richards
Víctor Robles
Paul Roe
Marta Sabou
Manuel Salvadores
Alberto Sánchez
Leon Sterling
Eleni Stroulia
Valentina Tamma
Henry Tirri
Santtu Toivonen
Chris van Aart
Julita Vassileva
Steve Willmott
Ning Zhong

CAMS 2005 Program Committee

Susanne Boll
Dan Chalmers
Keith Cheverst
Gill Dobbie
Tiong Goh
John Grundy
Dave Nichols
Goce Trajcevski
Mark van Setten
Agnes Voisard

GADA 2005 Program Committee

Jemal Abawajy
Akshai Aggarwal
Nedim Alpdemir
Mark Baker
Steve Benford
José Luis Bosque
Rajkumar Buyya
Mario Cannataro
Jesus Carretero
Elizabeth Chang
Steve Chiu
Toni Cortes
Vincenzo de Florio
María Eugenia de Pool
Stefan Egglestone
Alvaro Fernandes
Felix García
Chris Greenhalgh
Alastair Hampshire
Eduardo Huedo
Kostas Karasavvas
Daniel Katz
Domenico Laforenza
Ignacio Llorente
Phillip Lord
Bertram Ludaescher
Gianluca Moro
José María Peña
Omer Rana
Francisco Rosales
Rizos Sakellariou
Manuel Salvadores
Alberto Sánchez
Heinz Stockinger
Oliver Storz
Domenico Talia
David W. Walker
Laurence Yang

MIOS+INTEROP 2005 Program Committee

Antonia Albani
Giuseppe Berio
Bernhard Bauer
Christoph Bussler
Kamran Chatha
Emmanuel delaHostria
Jan L.G. Dietz
Joaquim Filipe
Rony G. Flatscher
Christian Huemer
Michael Huhns
Zahir Irani
Frank-Walter Jaekel
Peter Loos
Diego Milano
Michele Missikoff
Arturo Molina
Juergen Mueller
Hervé Panetto
Olivier Perrin
Colin Piddington
Gil Regev
Monica Scannapieco
Pnina Soffer
Arne Solvberg
Richard Stevens
Klaus Turowski
Vijay K. Vaishnavi
Bruno Vallespir
Alexander Verbraeck
René Wagenaar
Larry Whitman
Martin Zelm

ORM 2005 Program Committee

Scott Becker
Linda Bird
Anthony Bloesch
Peter Bollen
Andy Carver
Dave Cuyler
Aldo de Moor
Olga De Troyer
Necito dela Cruz
Jan Dietz
David Embley
Ken Evans
Gordon Everest
Henri Habrias
Pat Hallock
Terry Halpin
Stijn Hoppenbrouwers
Mustafa Jarrar
Alberto Laender
Robert Meersman
Tony Morgan
Sjir Nijssen
Erik Proper
Peretz Shoval
Sten Sundblad
Arthur ter Hofstede
Theo van der Weide
Gerd Wagner

PhDS 2005 Program Committee

Domenico Beneventano
Jaime Delgado
Jan L.G. Dietz
Werner Nutt

SeBGIS 2005 Program Committee

Gennady Adrienko
Yvan Bédard
Brandon Bennett
David Bennett
Michela Bertolotto
Alex Borgida
Christophe Claramunt
Eliseo Clementini
Nadine Cullot
Jean-Paul Donnay
Anders Friis-Christensen
Antony Galton
Werner Kuhn
Sergei Levashkin
Thérèse Libourel
Dimitris Papadias
Maurizio Rafanelli
Anne Ruas
Nectaria Tryfona
Peter van Oosterom
Stephan Winter

SWWS 2005 Program Committee

Aldo Gangemi
Amit Sheth
André Valente
Andrew Stranieri

Avigdor Gal
Carlos Sierra
Carole Goble
Carole Hafner
Cecilia Magnusson Sjoberg
Chris Bussler
David Bell
Elisa Bertino
Elizabeth Chang
Enrico Franconi
Ernesto Damiani
Feng Ling
Frank van Harmelen
Giancarlo Guizzardi
Grigoris Antoniou
Guiraude Lame
Hai Zhuge
Jaime Delgado
Jaiwei Han
John Debenham
John Mylopoulos
Joost Breuker
Jos Lehmann
Katia Sycara
Kokou Yetongnon
Layman Allen
Leonardo Lesmo
Ling Liu
Lizhu Zhou
Lotfi Zadeh
Manfred Hauswirth
Mariano Lopez
Masood Nikvesh
Mihaela Ulieru
Mohand-Said Hacid
Mukesh Mohania
Mustafa Jarrar
Nicola Guarino
Peter Spyns
Pieree Yves Schobbens
Qing Li
Radboud Winkels
Ramasamy Uthurusamy
Richard Benjamins
Rita Temmerman
Robert Meersman
Robert Tolksdorf
Said Tabet
Stefan Decker
Susan Urban
Tharam Dillon
Trevor Bench-Capon
Usuama Fayed
Valentina Tamma
Wil van der Aalst
York Sure

WOSE 2005 Program Committee

Lora Aroyo
Aldo de Moor
Erik Duval
Robert Farrell
Fabrizio Giorgini
Wayne Hodgins
Paul LeFrere
Ambjõm Naeve
Daniel Rehak
Tyde Richards
Peter Spyns
Frans Van Assche
Luc Vervenne
Martin Wolpers

Table of Contents

Posters of the 2005 CoopIS (Cooperative Information Systems) International Conference

Checking Workflow Schemas with Time Constraints Using Timed Automata
Elisabetta De Maria, Angelo Montanari, Marco Zantoni 1

Cooperation Between Utility IT Systems: Making Data and Applications Work Together
Claus Vetter, Thomas Werner 3

Adapting Rigidly Specified Workflow in the Absence of Detailed Ontology
Gregory Craske, Caspar Ryan 5

Modelling and Streaming Spatiotemporal Audio Data
Thomas Heimrich, Katrin Reichelt, Hendrik Rusch, Kai-Uwe Sattler, Thomas Schröder 7

Enhancing Project Management for Periodic Data Production Management
Anja Schanzenberger, Dave R. Lawrence, Thomas Kirsche 9

Cooperating Services in a Mobile Tourist Information System
Annika Hinze, George Buchanan 12

Posters of the 2005 DOA (Distributed Objects and Applications) International Conference

Flexible and Maintainable Contents Activities in Ubiquitous Environment
Kazutaka Matsuzaki, Nobukazu Yoshioka, Shinichi Honiden 14

Using Model-Driven and Aspect-Oriented Development to Support End-User Quality of Service
David Durand, Christophe Logé 16

A Generic Approach to Dependability in Overlay Networks
Barry Porter, Geoff Coulson 18

An XML-Based Cross-Language Framework
Arno Puder 20

Software Design of Electronic Interlocking System Based on Real-Time Object-Oriented Modeling Technique
Jong-Sun Kim, Ji-Yoon Yoo, Hack-Youp Noh 22

Posters of the 2005 ODBASE (Ontologies, Databases, and Applications of Semantics) International Conference

Ontology Based Negotiation Case Search System for the Resolution of Exceptions in Collaborative Production Planning
Chang Ouk Kim, Young Ho Cho, Jung Uk Yoon, Choon Jong Kwak, Yoon Ho Seo 24

Enhanced Workflow Models as a Tool for Judicial Practitioners
Jörn Freiheit, Susanne Münch, Hendrik Schöttle, Grozdana Sijanski, Fabrice Zangl 26

Semantics of Information Systems Outsourcing
H. Balsters, G.B. Huitema 28

Integration of Heterogeneous Knowledge Sources in the CALO Query Manager
José Luis Ambite, Vinay K. Chaudhri, Richard Fikes, Jessica Jenkins, Sunil Mishra, Maria Muslea, Tomas Uribe, Guizhen Yang 30

Context Knowledge Discovery in Ubiquitous Computing
Kim Anh Pham Ngoc, Young-Koo Lee, Sung-Young Lee 33

Ontology-Based Integration for Relational Data
Dejing Dou, Paea LePendu 35

Workshop on Agents, Web Services and Ontologies Merging (AWeSOMe)

AWeSOMe 2005 PC Co-chairs' Message 37

Document Flow Model: A Formal Notation for Modelling Asynchronous Web Services Composition
Jingtao Yang, Corina Cîrstea, Peter Henderson 39

Realising Personalised Web Service Composition Through Adaptive Replanning
Steffen Higel, David Lewis, Vincent Wade 49

Semantic Web Services Discovery in Multi-ontology Environment
Sasiporn Usanavasin, Shingo Takada, Norihisa Doi 59

Security and Semantics

On the Application of the Semantic Web Rule Language in the Definition of Policies for System Security Management
Félix J. García Clemente, Gregorio Martínez Pérez, Juan A. Botía Blaya, Antonio F. Gómez Skarmeta 69

On Secure Framework for Web Services in Untrusted Environment
Sylvia Encheva, Sharil Tumin 79

An Approach for Semantic Query Processing with UDDI
Jim Luo, Bruce Montrose, Myong Kang 89

Agents for Web Service Support

A Multiagent-Based Approach for Progressive Web Map Generation
Nafaâ Jabeur, Bernard Moulin 99

Semantics of Agent-Based Service Delegation and Alignment
H. Balsters, G.B. Huitema, N.B. Szirbik 109

Workshop on Context-Aware Mobile Systems (CAMS)

CAMS 2005 PC Co-chairs' Message 121

Personalising Context-Aware Applications
Karen Henricksen, Jadwiga Indulska 122

Management of Heterogeneous Profiles in Context-Aware Adaptive Information System
Roberto De Virgilio, Riccardo Torlone 132

Context-Aware Recommendations on the Mobile Web
Hong Joo Lee, Joon Yeon Choi, Sung Joo Park 142

Querying and Fetching

Capturing Context in Collaborative Profiles
Doris Jung, Annika Hinze 152

Using Context of a Mobile User to Prefetch Relevant Information
Holger Kirchner 156

Development and Engineering

Location-Based Mobile Querying in Peer-to-Peer Networks
Michel Scholl, Marie Thilliez, Agnès Voisard 166

Seamless Engineering of Location-Aware Services
Gustavo Rossi, Silvia Gordillo, Andrés Fortier 176

Location

Context-Aware Negotiation for Reconfigurable Resources with Handheld Devices
Timothy O'Sullivan, Richard Studdert 186

Location-Aware Web Service Architecture Using WLAN Positioning
Ulf Rerrer 196

A Light-Weight Framework for Location-Based Services
W. Schwinger, Ch. Grün, B. Pröll, W. Retschitzegger 206

Architecture and Models

Context Awareness for Music Information Retrieval Using JXTA Technology
Hyosook Jung, Seongbin Park 211

A Model of Pervasive Services for Service Composition
Caroline Funk, Christoph Kuhmünch, Christoph Niedermeier 215

Selection Using Non-symmetric Context Areas
Diane Lingrand, Stéphane Lavirotte, Jean-Yves Tigli 225

Sharing Context Information in Semantic Spaces
Reto Krummenacher, Jacek Kopecký, Thomas Strang 229

Grid Computing Workshop (GADA)

GADA 2005 PC Co-chairs' Message 233

Web Services Approach in the Grid

Coordinated Use of Globus Pre-WS and WS Resource Management Services with Grid*W*ay
Eduardo Huedo, Rubén S. Montero, Ignacio M. Llorente 234

Web-Services Based Modelling/Optimisation for Engineering Design
Ali Shaikh Ali, Omer F. Rana, Ian Parmee, Johnson Abraham, Mark Shackelford 244

Workflow Management System Based on Service Oriented Components for Grid Applications
Ju-Ho Choi, Yong-Won Kwon, So-Hyun Ryu, Chang-Sung Jeong 254

Grid Applications

Life Science Grid Middleware in a More Dynamic Environment
Milena Radenkovic, Bartosz Wietrzyk 264

A Grid-Aware Implementation for Providing Effective Feedback to On-Line Learning Groups
Santi Caballé, Claudi Paniagua, Fatos Xhafa, Thanasis Daradoumis 274

Security and Ubiquitous Computing

Caching OGSI Grid Service Data to Allow Disconnected State Retrieval
Alastair Hampshire, Chris Greenhalgh 284

Shelter from the Storm: Building a Safe Archive in a Hostile World
Jon MacLaren, Gabrielle Allen, Chirag Dekate, Dayong Huang, Andrei Hutanu, Chongjie Zhang 294

Event Broker Grids with Filtering, Aggregation, and Correlation for Wireless Sensor Data
Eiko Yoneki 304

Distributed Authentication in GRID5000
Sebastien Varrette, Sebastien Georget, Johan Montagnat, Jean-Louis Roch, Franck Leprevost 314

Performance Enhancement in the Grid

A Load Balance Methodology for Highly Compute-Intensive Applications on Grids Based on Computational Modeling
D.R. Martínez, J.L. Albín, J.C. Cabaleiro, T.F. Pena, F.F. Rivera 327

Providing Autonomic Features to a Data Grid
María S. Pérez, Alberto Sánchez, Ramiro Aparicio, Pilar Herrero, Manuel Salvadores 337

TCP Performance Enhancement Based on Virtual Receive Buffer with PID Control Mechanism
Byungchul Park, Eui-Nam Huh, Hyunseung Choo, Yoo-Jung Kim 347

Databases on the Grid

Computational Grid vs. Parallel Computer for Coarse-Grain Parallelization of Neural Networks Training
Volodymyr Turchenko 357

Object-Oriented Wrapper for Relational Databases in the Data Grid Architecture
Kamil Kuliberda, Jacek Wislicki, Radoslaw Adamus, Kazimierz Subieta 367

Modeling Object Views in Distributed Query Processing on the Grid
Krzysztof Kaczmarski, Piotr Habela, Hanna Kozankiewicz, Kazimierz Subieta 377

Optimization of Distributed Queries in Grid Via Caching
Piotr Cybula, Hanna Kozankiewicz, Krzysztof Stencel, Kazimierz Subieta 387

Replication

Modelling the $\sqrt{N}$ + ROWA Model Approach Inside the WS-ReplicationResource
Manuel Salvadores, Pilar Herrero, María S. Pérez, Alberto Sanchez 397

Workshop on Inter-organizational Systems and Interoperability of Enterprise Software and Applications (MIOS+INTEROP)

MIOS+INTEROP 2005 PC Co-chairs' Message 406

Service Modelling

Registering a Business Collaboration Model in Multiple Business Environments
Birgit Hofreiter, Christian Huemer 408

Component Oriented Design and Modeling of Cross-Enterprise Service Processes
Rainer Schmidt 421

Comparing the Impact of Service-Oriented and Object-Oriented Paradigms on the Structural Properties of Software
Mikhail Perepletchikov, Caspar Ryan, Keith Frampton 431

The Impact of Software Development Strategies on Project and Structural Software Attributes in SOA
Mikhail Perepletchikov, Caspar Ryan, Zahir Tari 442

Service Choreography and Orchestration

An Hybrid Intermediation Architectural Approach for Integrating Cross-Organizational Services
Giannis Verginadis, Panagiotis Gouvas, Gregoris Mentzas 452

A Framework Supporting Dynamic Workflow Interoperation
Jaeyong Shim, Myungjae Kwak, Dongsoo Han 461

A Text Mining Approach to Integrating Business Process Models and Governing Documents
Jon Espen Ingvaldsen, Jon Atle Gulla, Xiaomeng Su, Harald Rønneberg 473

A Process-Driven Inter-organizational Choreography Modeling System
Kwang-Hoon Kim 485

A Petri Net Based Approach for Process Model Driven Deduction of BPEL Code
Agnes Koschmider, Marco Mevius 495

From Inter-organizational Workflows to Process Execution: Generating BPEL from WS-CDL
Jan Mendling, Michael Hafner 506

PIT-P2M: ProjectIT Process and Project Metamodel
Paula Ventura Martins, Alberto Rodrigues da Silva 516

Interoperability of Networked Enterprise Applications

Requirements for Secure Logging of Decentralized Cross-Organizational Workflow Executions
Andreas Wombacher, Roel Wieringa, Wim Jonker, Predrag Knežević, Stanislav Pokraev 526

Access Control Model for Inter-organizational Grid Virtual Organizations
B. Nasser, R. Laborde, A. Benzekri, F. Barrère, M. Kamel 537

Interoperability Supported by Enterprise Modelling
Frank Walter Jaekel, Nicolas Perry, Cristina Campos, Kai Mertins, Ricardo Chalmeta 552

Using Ontologies for XML Data Cleaning
Diego Milano, Monica Scannapieco, Tiziana Catarci 562

Applying Patterns for Improving Subcontracting Management
Riikka Ahlgren, Jari Penttilä, Jouni Markkula 572

Evaluation of Strategic Supply Networks
Antonia Albani, Nikolaus Müssigmann 582

Self Modelling Knowledge Networks
Volker Derballa, Antonia Albani 592

Workshop on Object-Role Modeling (ORM)

ORM 2005 PC Co-chairs' Message 602

Schema Management

Using Abstractions to Facilitate Management of Large ORM Models and Ontologies
C. Maria Keet 603

Modularization and Automatic Composition of Object-Role Modeling (ORM) Schemes
Mustafa Jarrar 613

Modelling Context Information with ORM
Karen Henricksen, Jadwiga Indulska, Ted McFadden 626

Industry Perspectives

Using Object Role Modeling for Effective In-House Decision Support Systems
Eric John Pierson, Necito dela Cruz 636

Requirements Engineering with ORM
Ken Evans 646

Beyond Data Modeling

Generating Applications from Object Role Models
Betsy Pepels, Rinus Plasmeijer 656

A Fact-Oriented Approach to Activity Modeling
H.A. (Erik) Proper, S.J.B.A. Hoppenbrouwers, Th.P. van der Weide 666

Future Directions

ORM 2
Terry Halpin 676

A World Ontology Specification Language
Jan L.G. Dietz 688

Applications

Using ORM to Model Web Systems
Olga De Troyer, Sven Casteleyn, Peter Plessers 700

Object Role Modelling for Ontology Engineering in the DOGMA Framework
Peter Spyns 710

Formal Underpinnings

Fact Calculus: Using ORM and Lisa-D to Reason About Domains
S.J.B.A. Hoppenbrouwers, H.A. (Erik) Proper, Th.P. van der Weide 720

Schema Equivalence as a Counting Problem
H.A. (Erik) Proper, Th.P. van der Weide 730

Ph.D. Student Symposium

PhDS 2005 PC Co-chairs' Message 740

Accelerating Distributed New Product Development by Exploiting Information and Communication Technology
Darius Khodawandi 741

Towards QoS-Awareness of Context-Aware Mobile Applications and Services
Katarzyna Wac 751

Supporting the Developers of Context-Aware Mobile Telemedicine Applications
Tom Broens 761

Multilingual Semantic Web Services
Frédéric Hallot 771

Improving Information Retrieval Effectiveness by Using Domain Knowledge Stored in Ontologies
Gábor Nagypál 780

Top-k Skyline: A Unified Approach
Marlene Goncalves, María-Esther Vidal 790

Judicial Support Systems: Ideas for a Privacy Ontology-Based Case Analyzer
Yan Tang, Robert Meersman 800

IFIP WG 2.12 and WG 12.4 International Workshop on Web Semantics (SWWS)

SWWS 2005 PC Co-chairs' Message 808

Invited Papers on TRUST

Adding a Peer-to-Peer Trust Layer to Metadata Generators
Paolo Ceravolo, Ernesto Damiani, Marco Viviani 809

Building a Fuzzy Trust Network in Unsupervised Multi-agent Environments
Stefan Schmidt, Robert Steele, Tharam Dillon, Elizabeth Chang 816

Regulatory Ontologies (WORM)

Building e-Laws Ontology: New Approach
Ahmad Kayed 826

Generation of Standardised Rights Expressions from Contracts: An Ontology Approach?
Silvia Llorente, Jaime Delgado, Eva Rodríguez, Rubén Barrio, Isabella Longo, Franco Bixio 836

OPJK into PROTON: Legal Domain Ontology Integration into an Upper-Level Ontology
Núria Casellas, Mercedes Blázquez, Atanas Kiryakov, Pompeu Casanovas, Marta Poblet, Richard Benjamins 846

Applications of Semantic Web I (SWWS)

Semantic Transformation of Web Services
David Bell, Sergio de Cesare, Mark Lycett 856

Modeling Multi-party Web-Based Business Collaborations
Lai Xu, Sjaak Brinkkemper 866

A Framework for Task Retrieval in Task-Oriented Service Navigation System
Yusuke Fukazawa, Takefumi Naganuma, Kunihiro Fujii, Shoji Kurakake 876

Realizing Added Value with Semantic Web
Dariusz Kłeczek, Tomasz Jaśkowski, Rafał Małanij, Edyta Rowińska, Marcin Wasilewski 886

Applications of Semantic Web II (SWWS)

Ontology-Based Searching Framework for Digital Shapes
Riccardo Albertoni, Laura Papaleo, Marios Pitikakis, Francesco Robbiano, Michela Spagnuolo, George Vasilakis 896

Ontology Metadata Vocabulary and Applications
Jens Hartmann, Raúl Palma, York Sure, M. Carmen Suárez-Figueroa, Peter Haase, Asunción Gómez-Pérez, Rudi Studer 906

Ontological Foundation for Protein Data Models
Amandeep S. Sidhu, Tharam S. Dillon, Elizabeth Chang 916

Modeling and Querying Techniques for Semantic Web (SWWS)

SWQL – A Query Language for Data Integration Based on OWL
Patrick Lehti, Peter Fankhauser 926

Modeling Views for Semantic Web Using eXtensible Semantic (XSemantic) Nets
R. Rajugan, Elizabeth Chang, Ling Feng, Tharam S. Dillon 936

On the Cardinality of Schema Matching
Avigdor Gal 947

Reputation Ontology for Reputation Systems
Elizabeth Chang, Farookh Khadeer Hussain, Tharam Dillon 957

Ontologies (SWWS)

Translating XML Web Data into Ontologies
Yuan An, John Mylopoulos 967

Self-tuning Personalized Information Retrieval in an Ontology-Based Framework
Pablo Castells, Miriam Fernández, David Vallet, Phivos Mylonas, Yannis Avrithis 977

Detecting Ontology Change from Application Data Flows
Paolo Ceravolo, Ernesto Damiani 987

Workshop on Semantic-based Geographical Information Systems (SeBGIS)

SeBGIS 2005 PC Co-chairs' Message 997

Measuring, Evaluating and Enriching Semantics

How to Enrich the Semantics of Geospatial Databases by Properly Expressing 3D Objects in a Conceptual Model
Suzie Larrivée, Yvan Bédard, Jacynthe Pouliot 999

Evaluating Semantic Similarity Using GML in Geographic Information Systems
Fernando Ferri, Anna Formica, Patrizia Grifoni, Maurizio Rafanelli 1009

Measuring Semantic Differences Between Conceptualisations: The Portuguese Water Bodies Case – Does Education Matter?
Paulo Pires, Marco Painho, Werner Kuhn 1020

Schemata Integration

Spatio-temporal Schema Integration with Validation: A Practical Approach
A. Sotnykova, N. Cullot, C. Vangenot 1027

Preserving Semantics When Transforming Conceptual Spatio-temporal Schemas
Esteban Zimányi, Mohammed Minout 1037

Using Image Schemata to Represent Meaningful Spatial Configurations
Urs-Jakob Rüetschi, Sabine Timpf 1047

Geovisualization and Spatial Semantics

Collaborative geoVisualization: Object-Field Representations with Semantic and Uncertainty Information
Vlasios Voudouris, Jo Wood, Peter F Fisher 1056

Semantics of Collinearity Among Regions
Roland Billen, Eliseo Clementini 1066

Automatic Acquisition of Fuzzy Footprints
Steven Schockaert, Martine De Cock, Etienne E. Kerre 1077

The Double-Cross and the Generalization Concept as a Basis for Representing and Comparing Shapes of Polylines
Nico Van de Weghe, Guy De Tré, Bart Kuijpers, Philippe De Maeyer 1087

Algorithms and Data Structures

Range and Nearest Neighbor Query Processing for Mobile Clients
KwangJin Park, MoonBae Song, Ki-Sik Kong, Chong-Sun Hwang, Kwang-Sik Chung, SoonYoung Jung 1097

An Efficient Trajectory Index Structure for Moving Objects in Location-Based Services
Jae-Woo Chang, Jung-Ho Um, Wang-Chien Lee 1107

Systems and Tools

MECOSIG Adapted to the Design of Distributed GIS
Fabien Pasquasy, François Laplanche, Jean-Christophe Sainte, Jean-Paul Donnay 1117

The Emerge of Semantic Geoportals
Athanasis Nikolaos, Kalabokidis Kostas, Vaitis Michail, Soulakellis Nikolaos 1127

Ontology Assisted Decision Making – A Case Study in Trip Planning for Tourism
Eleni Tomai, Maria Spanaki, Poulicos Prastacos, Marinos Kavouras 1137

Workshop on Ontologies, Semantics and E-Learning (WOSE)

WOSE 2005 PC Co-chairs' Message 1147

E-Learning and Ontologies

Towards the Integration of Performance Support and e-Learning: Context-Aware Product Support Systems
Nikolaos Lagos, Rossitza M. Setchi, Stefan S. Dimov 1149

Taking Advantage of LOM Semantics for Supporting Lesson Authoring
Olivier Motelet, Nelson A. Baloian 1159

Repurposing Learning Object Components
Katrien Verbert, Jelena Jovanović, Dragan Gašević, Erik Duval 1169

Ontology Technology for E-learning

Interoperable E-Learning Ontologies Using Model Correspondences
Susanne Busse 1179

Towards Ontology-Guided Design of Learning Information Systems
Aldo de Moor 1190

Learning to Generate an Ontology-Based Nursing Care Plan by Virtual Collaboration
Woojin Paik, Eunmi Ham 1200

Ontologies and Virtual Reality

Generating and Evaluating Triples for Modelling a Virtual Environment
Marie-Laure Reinberger, Peter Spyns 1205

An Ontology-Driven Approach for Modeling Behavior in Virtual Environments
Bram Pellens, Olga De Troyer, Wesley Bille, Frederic Kleinermann, Raul Romero 1215

Author Index 1225

Checking Workflow Schemas with Time Constraints Using Timed Automata
(Extended Abstract)

Elisabetta De Maria, Angelo Montanari, and Marco Zantoni

Dipartimento di Matematica e Informatica, Università di Udine,
via delle Scienze 206, 33100 Udine, Italy
{demaria, montana, zantoni}@dimi.uniud.it

Abstract. Nowadays, the ability of providing an automated support to the management of business processes is widely recognized as a main competitive factor for companies. One of the most critical resources to deal with is time, but, unfortunately, the time management support offered by available workflow management systems is rather rudimentary. We focus our attention on the modeling and verification of workflows extended with time constraints. We propose (finite) *timed automata* as an effective tool to specify timed workflow schemas and to check their consistency. More precisely, we reduce the consistency problem for workflow schemas to the emptiness problem for timed automata, making it possible to exploit the machinery developed to solve the latter to address the former.

Workflow systems play an important role in the automation of business process management. They provide sophisticated tools for the specification and verification of the process structure (workflow schema), that allow one, for instance, to detect inconsistencies in process constraints and to identify process bottlenecks, as well as tools for monitoring and managing process execution, that make it possible, for instance, to trigger process-specific exception-handling activities when something wrong happens.

One of the most critical resources to deal with is time. Dealing with time constraints is crucial in designing and managing many business processes, and thus time management should be part of the core functionalities provided by workflow systems to control the lifecycle of processes. At build time, when workflow schemas are specified, workflow designers need means to represent time-related aspects of business processes, such as activity durations, time constraints between activities, deadlines, and timeouts, and to check their feasibility. At run time, when workflow schemas are instantiated and executed, process managers need mechanisms to detect possible time constraint violations and to trigger suitable exception-handling activities. Unfortunately, the time management support offered by available workflow management systems is rather rudimentary. The few existing approaches to time management in workflow systems are based on graph-theoretic techniques or on suitable refinements of Petri nets.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1–2, 2005.
© Springer-Verlag Berlin Heidelberg 2005

We propose (finite) *timed automata* as an effective tool to specify workflow schemas with time constraints and to check their consistency [1]. More precisely, we reduce the consistency problem for timed workflow schemas to the emptiness problem for timed automata, making it possible to exploit the machinery developed to solve the latter to address the former. Furthermore, we take advantage of tools for timed automata to solve other relevant problems, such as, for instance, the problem of checking whether there exists an execution of a consistent workflow that satisfies some specific temporal properties.

A workflow is a collection of activities, agents, and dependencies between activities. Activities can be executed sequentially, repeatedly in a loop, or in parallel. Parallel executions can be unconditional (all activities are executed), conditional (only activities that satisfy a certain condition are executed) or alternative (one activity among several alternative ones is executed). In addition, workflows may contain optional activities (some activities may be executed or not). The control structure of the workflow implicitly defines a number of structural time constraints that basically states that an activity can start only when its predecessor activities have been completed. Explicit time constraints can be added to take into account time restrictions on activities imposed by organizational rules, laws, and commitments. The most common explicit time constraints are those on the duration of activity execution and those that constrain the delay between pairs of activities. Timed automata are one of the most widely used tools for the specification and verification of real-time systems. They are obtained from classical ones by adding a finite set of real-valued clock variables (clocks for short). Constraints on clocks are added to both automata states and transitions. Decidability of the emptiness problem for timed automata can be obtained by imposing suitable restrictions to automata structure and/or clocks. We take advantage of the decidability of deterministic timed automata with clock constraints only comparing clock values with constants.

We first show how basic workflow constructs for activity composition can be rendered in terms of the automata operations of concatenation, union, product, and intersection. Then, we show how the explicit time constraints of a workflow schema (activity duration, relative deadlines, upper/lower bound constraints) can be encoded into constraints on the finite set of real-valued clocks of a timed automaton. Putting together these two ingredients, we define a translation algorithm that maps a large set of timed workflow schemas into the above-mentioned class of deterministic timed automata. In such a way, we reduce the problem of consistency checking for timed workflow schemas to a reachability problem for timed automata: we have that the constraints of a given timed workflow schema are satisfiable (consistency problem) if and only if the language recognized by the corresponding automaton is not empty (reachability problem).

References

[1] E. De Maria, A. Montanari, and M. Zantoni. Checking workflow schemas with time constraints using timed automata. Technical Report UDMI/06/05, University of Udine, Mathematics and Computer Science Dept., 2005.

Cooperation Between Utility IT Systems: Making Data and Applications Work Together

Claus Vetter and Thomas Werner

ABB Switzerland Ltd., Corporate Research, Segelhof, 5405 Baden-Daettwil, Switzerland
{claus.vetter, thomas.werner}@ch.abb.com

Abstract. The ongoing optimization of work processes requires a close cooperation of IT systems within an enterprise. Originating from requirements of the utility industry we present a concept of interoperability of utility software systems and its corresponding data. Our solution builds on industry standards - Common Information Model (CIM) as a power system domain data model and SOAP as a standard for messaging and interface specification. Together they provide a basis for translating data between applications and are seamlessly bound to a communication infrastructure.

1 Introduction

Under the pressure of market liberalization in e.g., energy markets, high maintenance costs for IT systems and a growing need to share data across applications enforce a harmonization of the underlying IT infrastructure. Electric utilities are companies which generate, distribute and handle power, maintain an electrical network and supply customers with electrical energy. The utilities' operations business can be divided into network control, which uses a SCADA system (Supervisory Control and Data Acquisition) to control and operate the electricity grid; network maintenance, which uses a CMMS (Computerized Maintenance Management System) to maintain the electrical network and its equipment; and network planning which is responsible for managing extensions of the electric network. These utility software systems - from a data modeling perspective - very often share information which resides in more than one application, although dealing with the same physical assets, such as transformers, breakers or generators. To enable collaboration between the IT systems, this information must now be shared between disparate software systems throughout the enterprise. SCADA, CMMS and network planning systems are each implementing their own information model different with respect to syntax, semantics and granularity of the information. From the viewpoint of the modeled asset itself however, these applications share to a significant extent the same data.

Our solution is able to deal with and translate multiple data models, allow for changes in data attributes, allow for insertion and removal of data objects and be extensible for future changes in the data representation. The main communication requirement is that the infrastructure must be able to guarantee the delivery of messages (instructions and data) when an application requires so. In fact the communication must be secured with respect to the data being delivered and in the right delivery order. The more, many customers already have specific communication infrastruc-

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 3 – 4, 2005.
© Springer-Verlag Berlin Heidelberg 2005

tures deployed that can place further constraints upon the communication channel used. As a main non-functional requirement on both data and communication, more and more focus is put on adherence to standards.

2 The Solution Concepts

We apply the Common Information Model (IEC TC57) which defines a common semantic (data object and attributes) understandable by all applications through translation mechanisms. In order to manage the data translations and the references between the various applications and the global representation we introduce a directory service. This service holds the mapping information that applications need for mediating between data models. Adapters act as "glue" to link applications to the communication layer by transforming information between individual and global information models and providing means to access the applications as well as means to publish information to the outside. An adapter typically consists of three main building blocks, data access functionality to read information from and write information to the applications including data transformations; address changes, such as insertion and deletion of objects and ensuring the consistency of reference containers hosted by the directory service and notifications on changes for objects and their attributes from the applications. SOAP as a messaging protocol of the communication layer allows us to alter services and add or remove applications without affecting the core interaction of the remaining components. This enables us to innovate how these services are delivered without affecting the other parts of the solution. The actual transport protocol is chosen during the instantiation of the service, and can be changed during runtime. Each transport binding (e.g., tcp:// or msmq://) determines the communication stack which is used for talking to the wire.

3 Conclusions and Summary

We show that integration solutions for electric utilities are confronted with challenges that originate from heterogeneous information models and a wide area of technologies. On the data side, the main benefit of the presented approach is the abstraction on a type based standard data model. We overcome technology heterogeneousness by decoupling application functionality into services and using SOAP as a vendor-neutral communication protocol.

Adapting Rigidly Specified Workflow in the Absence of Detailed Ontology

Gregory Craske and Caspar Ryan

School of Computer Science and Information Technology,
RMIT University, Melbourne, Australia
{craske, caspar}@cs.rmit.edu.au

Introduction and Rationale

Adaptive workflow approaches (for example, [1]) promise to provide flexibility of web service composition. However, definition-time adaptive workflow approaches (for example, exception handlers, and alternative flow selection) do not account for service environment dynamics, such as availability of new services and changing QoS parameters of services. This paper introduces a new method of automatic, run-time adaption, called workflow *convergence*. It utilises ontology in early development; ontology that reflects service message structures though is not semantically rich enough to support pure semantics based discovery and composition [2,3]. Industry is yet to widely embark on developing the complex semantic models that are fundamental for these approaches. Our workflow adaptation approach ensures that ontology is useful and value-added at *all* stages of development, thus providing an *added incentive for industry to adopt such ontology modelling efforts*. The convergence approach is introduced in the following section. Convergence relies on a service description approach, introduced in the last section, that utilises ontology in early development.

Workflow Convergence

Convergence is a procedure in which tasks in a workflow are merged to create new tasks in the place of the merged ones. The purpose of this is to allow a workflow to adapt to changes in the service environment that mandate a task replacement. For example, this could be due to a partner service no longer satisfying business or technical constraints by becoming faulty or no longer satisfying QoS constraints, thus, a new service must be found to replace it. If no service can directly replace the failed one, then tasks can be merged to accommodate a courser grained service that provides both the functionalities of the merged nodes. Thus, convergence always results in a smaller workflow topology, though its function is preserved.

Figure 1 uses a Petri-net workflow representation to show how convergence affects workflow design patterns [4]. Figure 1.a shows two parallel flow sequences where a task failure results in a convergence on tasks t_1 and t_3. Thus, $t_{1,3}$ becomes a synchronisation point providing a combined functionality, and does not

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 5–6, 2005.
© Springer-Verlag Berlin Heidelberg 2005

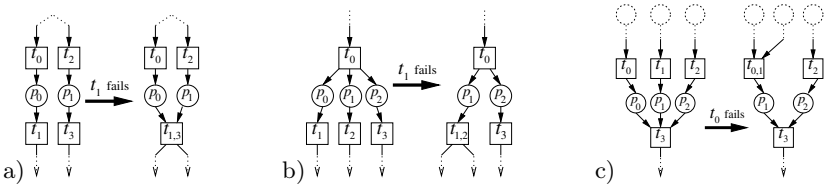


Fig. 1. Converged workflow patterns: a) sequence, b) parallel split, c) synchronisation

introduce any flow anomalies. Similar for Figure 1.b and Figure 1.c, where such convergence is the case within the parallel split and synchronisation patterns respectively.

ME-nets

In practice, convergence is governed by correctness criteria that are based on both the context of service inclusion in the workflow, and the functional requirements of each service. For this, a service interface representation called Message Event nets (ME-nets) has been proposed. ME-nets are Petri-net notations for specifying the data flow through the service. Each place in an ME-net can have an associated message type to show the message structures of the data being passed between operations of the service. These types can be associated with an ontology that represents structural and semantic relationships between these types as the foundation of a broader, richer domain ontology. A workflow can have two services connected in a data flow, and the place that connects them can take on a message type appropriate to that data flow. Future work focuses on service discovery, selection, and incorporation mechanisms that guarantee workflow functional equivalence before and after convergence.

Acknowledgement. This project is fully funded by the ARC (Australian Research Council), under Linkage scheme no. LP0347217 and SUN Microsystems.

References

1. Tang, J.F., Zhou, B., He, Z.J., Pompe, U.: Adaptive workflow-oriented services composition and allocation in distributed environment. In: Proceedings of International Conference on Machine Learning and Cybernetics, Vol.1. (2004) 599–603
2. Patel, C., Supekar, K., Lee, Y.: Provisioning resilient, adaptive web services-based workflow: a semantic modeling approach. In: Proceedings of the IEEE International Conference on Web Services. (2004) 480–487
3. Buhler, P., Vidal, J.M.: Towards adaptive workflow enactment using multiagent systems. Information Technology and Management Journal **6** (2005) 61–87
4. van der Aalst, W., ter Hofstede, A., Kiepuszewski, B., Barros, A.: Workflow patterns. Distributed and Parallel Databases **14** (2003) 5–51

Modelling and Streaming Spatiotemporal Audio Data

Thomas Heimrich[1], Katrin Reichelt[2], Hendrik Rusch[1],
Kai-Uwe Sattler[1], and Thomas Schröder[1]

[1] Technical University of Ilmenau,
Department of Computer Science and Automation, Germany
[2] Fraunhofer Institute for Digital Media Technology IDMT, Germany

1 Workflow of Sound Production

In this paper[1], we describe a special application domain of data management – the production of high-quality spatial sound. The IOSONO system[2], developed by Fraunhofer IDMT, is based on the wave field synthesis. Here, a large number of loudspeakers is installed around the listening room. A rendering component computes the signal for each individual speaker from the position of the audio source in a scene and the characteristics of the listening room. So, we can achieve the impression that sound sources are on specific positions in the listening room.

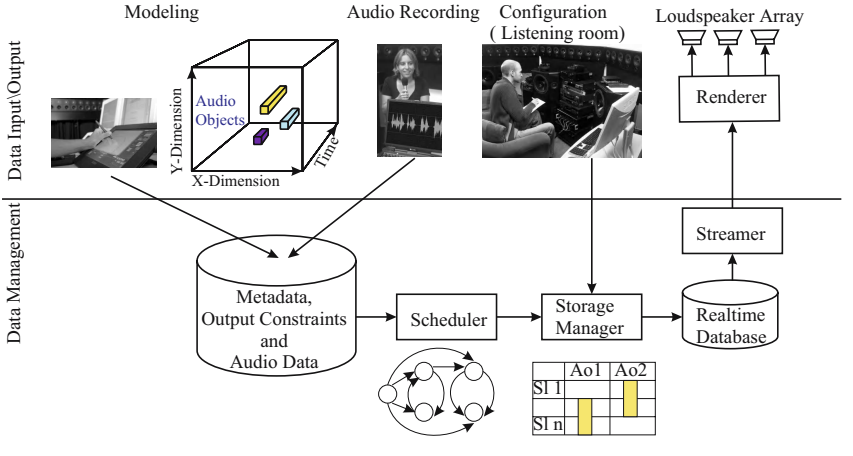


Fig. 1. Workflow of Sound Production

We distinguish two main phases of sound production and streaming (Fig. 1). In the *modeling* phase an audio scene is composed from audio objects in a cooperative work process. Each audio object is parametrized by properties such

[1] An extended version of this paper can be found at http://www3.tu-ilmenau.de/site/ipim/Publikationen.180.0.html
[2] www.iosono-sound.com

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 7–8, 2005.
© Springer-Verlag Berlin Heidelberg 2005

as position and loudness. This scene description together with the actual audio data is the input for the *rendering* phase.

2 Modelling Audio Scenes

For an efficient and comfortable modelling of audio scenes we need a comfortable graphical user interface. Furthermore, a relative modelling of temporal and spatial relations between sound sources must be possible. Special *output constraints* should be used to model these relationships. Output constraints use the well known Allen-Relations for a descriptive definition of temporal and spatial relationships between audio sources. Output constraints can be supported by a database system. So, it can check whether a set of output constraints is consistent or not. Furthermore, it can guarantee consistency between output constraints and audio data stored in the database.

3 Managing Data of Audio Scenes

We chose the Berkeley DB storage manager[3] as the platform for our data management component. Audio scenes are stored in a special XML structure.

Version Control supports the cooperative sound production process. The versioning subsystem implemented is based on version numbers. For each attribute a valid time, represented by start and end version number, is stored.

Searching audio objects and metadata is supported by XQueries which can be implemented easily using Berkeley DB.

Output Constraints Checking must be done efficiently during data input and output. Therefore, we need an adequate representation of these constraints. So, we transform output constraints in special inequalities called *difference constraints* which can be checked efficiently.

4 Data Organization for Realtime Audio Streaming

For rendering we need a reading rate for audio data up to 11 MBit/s. To allow such an output rate, we have developed a *special file structure* for storing audio objects. The idea is to generate a physical data organization from the output schedules built by the output scheduler (figure 1). Output schedules are logical orders for the temporal and spatial output of sound sources built from output constraints. By using output schedules we can store audio data in playtime order, thus at runtime the file can usually be chronologically read without jumps in either direction.

A detailed evaluation of the developed data structure as well as standard database structures can be found in the extended version of this paper.

[3] www.sleepycat.com

Enhancing Project Management for Periodic Data Production Management

Anja Schanzenberger[1,2], Dave R. Lawrence[2], and Thomas Kirsche[1]

[1] GfK Marketing Services, Nordwestring 101, 90319 Nuremberg, Germany
{Anja.Schanzenberger, Thomas.Kirsche}@gfk.de
[2] Middlesex University, Trent Park, London, N11 2NQ, United-Kingdom
{A.Schanzenberger, D.Lawrence}@mdx.ac.uk

Abstract. When data itself is the product, the management of data production is quite different from traditional goods production management. Production status, the quality of the products, product identifiers, deviations, and due dates are defined in terms of volatile data and are handled strictly to enable the resulting reports within the allotted time. This paper outlines how the information gathering process for a data production management system can be automated. The system's architecture is based upon ideas of project management. Milestones are enriched with production information. The major benefits are the following. Operators understand easily this management. They can concentrate on production itself, but are provided with reliable management information without manual effort. Additionally, with this solution a production plan is automatically created in advance.

1 Introduction

Data production systems [1] specialise in the analysis and transformation of large data quantities. The production consists usually of a periodically repeated workflow. Data packages, the product parts, flow through this workflow. The end product is 'information' in form of statistical reports. Data production management means to control timing, costs and resources and includes planning, monitoring and controlling data production [3]. We focus here on timing. Requirements in data production management are the following issues: Showing the status of data packages throughout the workflow. Providing quality means to avoid production errors and to obtain an optimum of data packages scheduled within the allotted time. Overcoming data aggregations and segmentations. Coping with unstable data identifiers as data packages change their identification keys during production [3]. Handling the frequent deviations at run-time [3]. Using exception reporting for management information reduction. Using the periodic repetition for automating the planning. Concentrating on progress monitoring rather than direct corrections in production. In this paper the loosely coupled management approach we introduced in [3] is used. Its advantage is production is not coupled to its management system. Instead only the production progress is queried. Data production is fully independent. Here we sketch how the information gathering for supplying the management system can be automated.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 9 – 11, 2005.
© Springer-Verlag Berlin Heidelberg 2005

2 Automated Milestone Creation in Data Production Management

GfK Marketing Services [2], a leading market researcher, has established a world-wide distributed, component-based data production system as sketched in [3]. Here we outline how we introduced in this case-study background processes to automate data production management. As the approach in [3] shows milestones have to be enriched with data content information and progress degrees. Checkpoints represent the different points of interest in the production process and are templates for milestones. Predecessor-successor relationships enable the display of relationships between milestones. In the case of our approach, milestones are automatically created after 'start triggering'. The following building blocks are needed to achieve this:

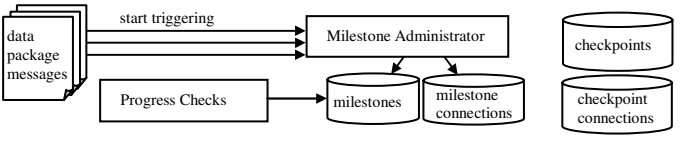


Fig. 1. Software architecture of the milestone module

Milestone Administrator: For each message of new incoming data packages this program creates all necessary milestones at the checkpoints and all connections between them, by quering production (see figure 1). Even creating milestones for the future is possible, as due dates can be estimated from former production periods. Operators can also register 'due date rules' to control the planning. These rules are considered in the milstone administrator. This enables operators to plan production in advance.

Milestone Progress Checks: In regular intervals a program checks the status and progress of milestones.

User Interface: A web-based user interface is provided for the milestone management. By using the predecessor and successor relationships, the user can navigate through the whole production process.

3 Conclusions and Research Plan

Our contribution here is to sketch for a data production management system how the processes for gathering the management information can be fully automated. The benefit is operators can focus on their daily business without interruption, but are provided with reliable management information. This solution is able to automatically create a production plan in advance. The user interface offers querying on problem cases like production delays. This has been proven in a real-world case-study. We expect to provide further management reports like Gantt and Pert diagrams, critical path methods and reports of due date adherence, adjusted for data production.

References

1. Albrecht, J., Lehner, W., Teschke, M., Kirsche, T.: Building a Real Data Warehouse for Market Research, Proc. DEXA 1997 Workshop, p.651-656
2. GfK Marketing Services: [Online], http://www.gfkms.com, [2005, Aug., 15]
3. Schanzenberger, A., Lawrence, D.R.: Automated Supervision of Data Production – Managing the Creation of Statistical Reports on Periodic Data., in Meersman, R., Tari, Z. (Eds.): Proc. CoopIS/DOA/ODBASE Conf. 2004., Lecture Notes in Computer Science, Vol. 3290, Springer Verlag, Berlin Heidelberg New York (2004)., p.194-208

Cooperating Services in a Mobile Tourist Information System

Annika Hinze[1] and George Buchanan[2]

[1] University of Waikato, Hamilton, New Zealand
a.hinze@cs.waikato.ac.nz
[2] UCL Interaction Centre, London, United Kingdom
g.buchanan@cs.ucl.ac.uk

1 Introduction

Complex information systems are increasingly required to support the flexible delivery of information to mobile devices. Studies of these devices in use have demonstrated that the information displayed to the user must be limited in size, focussed in content [1] and adaptable to the user's needs [2]. Furthermore, the presented information is often dynamic – even changing continuously. Event-based communication provides strong support for selecting relevant information for dynamic information delivery.

We report about the extension of our mobile tourist information system TIP with cooperating services. We are building upon our first-generation stationary core system, TIP 1.0 [3]. Previous work focussed on the interplay of different event/information sources and the event-based information delivery.

The extended TIP architecture provides users with information from modular, cooperating, mobile services. Modular services can be used in addition to the core system, allowing the users to use different services for similar purposes interchangeably (e.g., for guidance information in maps or textual representation). Cooperating services exchange context data and information for the benefit of the system's users. Mobile services can be used on typical hand-held devices; preferably with little or no installation and maintenance overhead for the user.

2 Issues Identified and Lessons Learned

We have identified a number of difficulties and challenges for creating an event-based communication framework for mobile systems.

Communication Protocols. The range of inter-process communication techniques that are available between processes running on the same mobile device are limited and rather restrictive. On the other hand, communication between a mobile device and the TIP server can have several forms. A global framework should to hide such implementation details from the different components of the system.

Service Composition. The final service provided to the user (e.g., a tourist guide with map, sight data and recommendations) is internally a composition of a variety of services. Services need to communicate both within the same

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 12–13, 2005.
© Springer-Verlag Berlin Heidelberg 2005

machine and between computers, using thick- and thin-client scenarios, or hybrid approaches. A sound framework must support new types of services and alternative realizations of the same service.

Standards. We have created a framework in which mobile services cooperate, but further forms of co-operation and composition are needed, e.g., to support new inter-process communications. Implementation details, particularly issues of standardisation, continue to be relevant. In the map system, different mapping scales and notations are used by different map and information providers, and further services must be introduced to mediate between systems that function in different notational standards.

As information systems move onto mobile devices or support mobile clients, the challenges identified here will become more pronounced. Client devices will provide a number of pre-installed services and users will add their own selections. Consequently, even stronger decoupling and modularization may be needed: A mobile infrastructure for mobile information services needs to flexibly support existing, changing or new services. The next design step in the TIP project will therefore see the completion of re-designing TIP into a Service-Oriented Architecture (SOA) using web services (TIP 3.0).

3 Conclusion

This paper discussed services in our TIP 2.9 prototype of a mobile tourist information system. TIP provides a new mobile infrastructure for cooperating information services, based on an event-based communication layer to support continually changing information. No existing systems fully address the problems of modular incorporation of and cooperation between various services in a mobile information delivery system.

In future work, we wish to extend the cooperation (and thus communication) between the provided services. We also plan to incorporate new services, such as access to external information sources, e.g., in digital libraries. This may lead to further exploration of sophisticated context-models which can be used for standardized communication between the services. We wish to support even more flexible service utilization: services may register and unregister depending on availability and capability of the mobile device. For TIP 3.0, we will employ a Service-Oriented Architecture (SOA) using web services.

References

1. G Buchanan and M Jones. Search interfaces for handheld web browsers. In *Poster Proceedings of the 9th World Wide Web Conference, Amsterdam, Netherlands*, September 2000.
2. Dina Goren-Bar. Overcoming mobile device limitations through adaptive information retrieval. *Applied Artificial Intelligence*, 18(6):513–532, 2004.
3. A. Hinze and A. Voisard. Location- and time-based information delivery in tourism. In *Conference in Advances in Spatial and Temporal Databases (SSTD 2003)*, volume 2750 of *LNCS*, Greece, July 2003.

Flexible and Maintainable Contents Activities in Ubiquitous Environment

Kazutaka Matsuzaki[1], Nobukazu Yoshioka[2], and Shinichi Honiden[1,2]

[1] University of Tokyo, 7-3-1 Hongo, Bunkyo-ku,
Tokyo 113-0033, Japan
[2] National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku,
Tokyo 101-8430, Japan
{matsuzaki, nobukazu, honiden}@nii.ac.jp

Abstract. In future ubiquitous environments, contents (data, movie, text, graphics, etc.) will be more sophisticated and context-aware so that they can enrich user experience. We have proposed the Active Contents (AC) framework, which is based on contents encapsulation with program and aspect definition to allow contents to behave actively. AC can be seen as a software component with several viewpoints of contributors (planner, designer, programmer, etc.). The problem is about maintainability of AC which is modified by the contributors based on their own viewpoints.

In this position paper, we propose a mechanism to allow such contributors to modify AC with context-aware aspect. In our mechanism, based on location binding analysis for AC, parallel executions to be performed at a separate location are detected and automatically executed using workflow-aware communication.

1 Introduction: Active Contents

In future ubiquitous environment, more and more contents would prevail in the network and people would use surrounding service appliances much easier. Contents will be allowed to act with context-awareness to make use of such services. Raw contents are encapsulated with programs (workflow)[1] and aspects using mobile agent techniques, which is named Active Contents (AC) [1]. Contents will autonomously perform workflow even after leaving their contributors hands.

As for development of AC, there are several contributor roles that will modify the program of AC according to their viewpoints. This modification falls into bad maintainability of the program. In order to avoid this, we apply Aspect-Oriented Programming for extension of behavior of AC.

2 Internal/External Context-Aware Active Contents: Workflow-Aware and Location-Aware Model

We have developed the AC framework which has a context-aware aspect interpretation part and a component repository (Fig. 1). The AC framework receives

[1] Process driven-modeling is assumed. A workflow contains several activities which is a unit of execution (e.g. Java Runnable class).

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 14–15, 2005.
© Springer-Verlag Berlin Heidelberg 2005

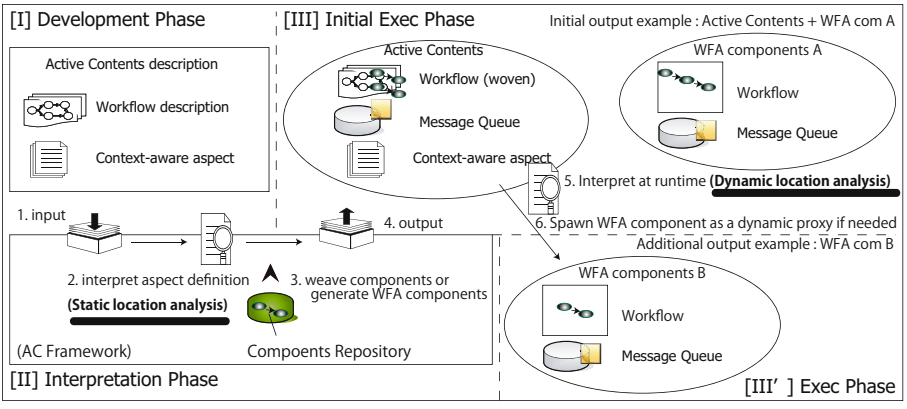


Fig. 1. Interpretation model of AC. AC descriptions (workflow, aspect) are inputted [I] and location inconsistencies are analyzed. Aspect components are woven into either the original workflow or a WFA component depending on the analysis [II]. Even during the execution period, dynamic location analyses are performed and other WFA components could be spawned [III, III'].

inputs (descriptions of workflow and aspects for an AC) and performs analysis of location bindings (e.g. execute on user's mobile device) of each activity in the workflow. The framework appends the workflow with extensive activities from component repository based on the aspect description. If an inconsistency of location bindings is found (i.e. to be performed parallel at separate hosts), a WFA (WorkFlow-Aware) component is deployed to solve them. The location analyses are performed even during the execution period and dynamically launch other WFA components when needed. These WFA components are allowed to watch the progress and some data of the original workflow of the AC through automatic message passing (WFA messaging). With this mechanism, aspect can be internal-context-aware (i.e. WFA) and insert actions at appropriate points without disrupting the original workflow, which leads to maintainability of AC.

References

1. Kiczales, G., Lamping, J., Menhdhekar, A., Maeda, C., Lopes, C., Loingtier, J.M., Irwin, J.: Aspect-oriented programming. In: Object-Oriented Programming 11th European Conference. (1997)

Using Model-Driven and Aspect-Oriented Development to Support End-User Quality of Service

David Durand and Christophe Logé

Laria, UPJV, 33 rue du Saint-Leu, 80039 Amiens, France
{david.durand, christophe.loge}@u-picardie.fr

1 Introduction

Nowadays, more and more applications are distributed and require runtime guarantees from their underlying environment. To reach theses guarantees, Quality of Service (QoS) information must be provided. The development of QoS-aware applications requires the control and management of low-level resources. It is a complex task to correlate high-level domain-specific QoS requirements with low-level system-dependent characteristics. Various middleware architectures have been proposed to provide a QoS support [1][2] and to free the designer from low-level resource concerns. Some architectures [3] use dynamic adaptation of applications or a component-based conception [4]. Our approach uses the Model Driven Architecture [5]: the mapping of QoS constraints with platform specific resource management can be automated during the transformation steps leading from the model to the software implementation. In this paper, we present a solution to model high-level user-oriented QoS constraints, and we demonstrate that the MDA, associated with Aspect-Oriented Software Development, can ease the conception of QoS aware application.

2 QoS Specification and Transformation in the MDA Process

In order to describe meta-information closer to end-user needs, we propose to model QoS Specifications at a resource-independent and platform-neutral abstraction level. We also propose mechanisms to translate high-level information to lower level parameters. Using the OMG's QoS Profile metamodel [6], we define a set of properties to build a profile adapted to the specification of high-level requirements from the end-user viewpoint [7]. The management and control of underlying system resources are hidden and reappear after the automatic transformation of end-user parameters. Our objective is to get software bundles that contain all the necessary code to execute the application with embedded QoS management in distributed environment. To achieve this goal, we integrate middleware architectures, COTS resource-management components, and code instrumentation technologies. In a first time, we extract the elements of the model stereotyped as QoS-managed. In a second time, we identify the platform-specific parameters according to the application designer guidelines. Several factors determinate the sequence of steps involved in the application process: (i) the target middleware architecture and operating system; (ii) the programming language chosen to develop the application; (iii) instrumentation technology to inject management code; (iv) COTS resource management components available for the forenamed parameters.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 16 – 17, 2005.
© Springer-Verlag Berlin Heidelberg 2005

To ensure the QoS enforcement according to user-defined constraints in the model, we define low-level characteristics mappings and supply their values for each application domain and quality levels. These parameters are intended to the runtime components that handle QoS requirement verifications and resources management.

Code generation represents the last transformation step in MDA. After the gathering of all information about the selected target environment, we produce the application structure and configuration files needed for the build of the final software.

Aspect Oriented Programming [8] is used at this step to inserts hooks into the code that link the application logic with the QoS Manager. In the resulting application bundle, instrumented code informs the manager about the application's activity: method calls, object lifecycle, data transmission or modification. QoS management code is thus centralized in an external module whose role is to manage and control resource, to verify the QoS rules and to apply the specified transition policies.

3 Conclusion

We outlined the application of the MDA principles for the conception of software that need quality of service guarantees. Our approach differs from common available QoS frameworks, in that the relation between high-level QoS specification and low-level resources is strongly decoupled. As far as QoS management is externalized from the application code, we use AOP as an instrumentation technology to link application components to QoS management components. This principle eases the maintenance and helps to focus on the business logic without dealing with technical low-level functionalities that require specialized skills.

References

1. Schmidt D.C., Levine D.L., Mungee S., "The Design and Performance of Real-Time Object Request Brokers", Computer Communications, vol. 21, pp. 294-324, 1998.
2. Li B., "Agilos: A Middleware Control Architecture for Application-Aware Quality of Service Adaptations", PhD Dissertation, University of Illinois at Urbana-Champaign, 2000.
3. Truyen E., Joergensen B.N., Joosen W., "Customization of Object Request Brokers through Dynamic Reconfiguration", Tools Europe 2000, Mont-St-Michel, France, 2000.
4. Wang N, Balasubramanian K., Gill C., "Towards a Real-time CORBA Component Model", OMG Workshop On Embedded & Real-Time Distributed Object Systems, 2002.
5. Object Management Group, "MDA Guide v1.0", document omg/03-06-01, 2001.
6. Object Management Group, "UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics", document ptc/04-06-01, 2004.
7. Durand D., Logé C., "End-User Specification of Quality of Service : Applying the Model-Driven Approach", joint ICAS & ICNS, Papeete, French Polynesia, 2005.
8. Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Videira Lopes, C., Loingtier, J.-M., and Irwin, J. "Aspect-Oriented Programming", ECOOP, Jyväskylä, Finland, 1997.

A Generic Approach to Dependability in Overlay Networks

Barry Porter and Geoff Coulson

Computing Department, Lancaster University, Lancaster, UK
(barry.porter, geoff)@comp.lancs.ac.uk

Overlay networks are virtual communication structures that are logically "laid over" underlying hosting networks such as the Internet. They are implemented by deploying application-level topology maintenance and routing functionality at strategic places in the hosting network [1,2]. In terms of *dependability*, most overlays offer proprietary "self-repair" functionality to recover from situations in which their nodes crash or are unexpectedly deleted. This functionality is typically orthogonal to the purpose of the overlay, and a systematic and complete approach to dependability is rarely taken because it is not the focus of the work. We therefore propose to offer dependability as a *service* to any overlay.

Dependability is a well-studied field in distributed applications, but many of the existing approaches for applications are unsuitable for overlays; a common approach is to submit an application to a fault-tolerant framework for controlled execution within that framework [3]. We suggest instead that a dependability service for overlays is built only from decentralized, lightweight agents which exist *alongside* overlay nodes, operating at the same level and with the same resources available to their nodes. In addition, such agents should maintain only soft state which can be re-built automatically simply by existing in the environment, making the service inherently self-repairing. An architectural model of this is shown in figure 1.

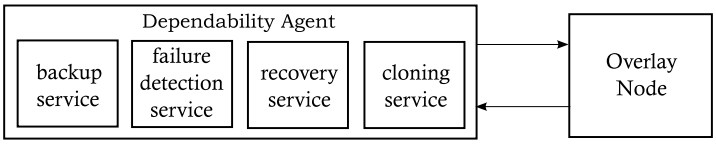


Fig. 1. An example configuration of the proposed overlay dependability service

This decentralization creates some interesting challenges, not least of all that recovery of a failed node can be initiated by multiple different instances of the recovery service that notice the failure of a neighbouring node. The service instances must then ensure that exactly one proposed recovery is chosen from potentially many in order to maintain a sensible system. We also note that overlays often operate in a purely end-user host environment, which can be quite limited in resources (or at least resources that users are happy to give up). This

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 18–19, 2005.
© Springer-Verlag Berlin Heidelberg 2005

makes some classic approaches infeasible, like the use of dedicated checkpointing servers. Checkpointing is a well-known method of making elements of a distributed system survivable—their state is periodically saved on dedicated servers so that it can be restored if they fail. Without the ability to provision checkpointing servers, checkpoints must be stored on the hosts used by the overlay itself, and must be distributed across those hosts in such a way that multiple failures are survivable.

Checkpointing itself, though often used in distributed applications, is not very suitable for overlays. Checkpointing is best used in deterministic systems, a class that many overlays are not part of—the data stored by Chord and the members of a multicast tree are both driven by users, for example, which makes those aspects appear random. Since overlays can, then, be in a near-constant state of non-determinism, their nodes would require frequent checkpointing, which can amount to a significant performance overhead.

To help alleviate issues like this we propose that overlays be loosely defined by two basic types from the point of view of the dependability service, which we term *accessinfo* and *nodestate* records; the former provides the service with the neighbours of a node to both save and communicate through, and the latter gets any part of a node that needs to be backed up (both are defined by the overlay and their internals are transparent to the service). If supported by the overlay, nodestate records may themselves be divisible into nodestate *units* which could, for example, map to individual resources stored at a node, again as determined by the overlay. This finer-grained abstraction gives us opportunities to back up only the elements of a node which change.

We are also interested in the performance of overlays when they are operating across highly heterogeneous hosts, as many overlays assign equal responsibility to each node regardless of the capabilities of its host. Re-using our definition, we can migrate nodestate units to "cloned" versions of their original nodes instantiated on alternative hosts, re-routing messages appropriately, to help alleviate pressure on the more sparsely resourced hosts that are part of an overlay. This behaviour is encapsulated in our cloning service, which monitors resources and attempts to prevent resource exhaustion by migrating and tracking nodestate units.

We intend to investigate solutions to all of the presented problems and expand on our loose overlay definition in our effort to create a generic, configurable and efficient dependability service for overlay networks.

References

1. Mathy, L., Canonico, R., Hutchinson, D.: An Overlay Tree Building Control Protocol. Proceedings of the 3rd International COST264 Workshop on Networked Group Communication, London, UK, 2001
2. Stoica, I. et al: Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. Proceedings of SIGCOMM'01, California, USA, 2001
3. Bagchi, S., Whisnant, K., Kalbarczyk, Z., Iyer, R.: The Chameleon Infrastructure for Adaptive, Software Implemented Fault Tolerance. Proceedings of the Symposium on Reliable Distributed Systems, Indiana, USA, 1998

An XML-Based Cross-Language Framework

Arno Puder

San Francisco State University, Computer Science Department,
1600 Holloway Avenue, San Francisco, CA 94132
arno@sfsu.edu

Abstract. We introduce XMLVM, a Turing complete XML-based programming language based on a stack-based, virtual machine. We show how XMLVM can automatically be created from Java class-files and .NET's Intermediate Language. While the programmer is never directly exposed to XMLVM, we provide tools based on XMLVM for tasks such as cross-language functional testing or code migration.

Overview

XMLVM is a Turing complete, fine-granular XML-based programming language. XMLVM uses a generalized virtual machine model that allows us to translate Java byte code as well as .NET executables (which are also based on a virtual machine concept) to XMLVM. We therefore manage to embrace both the Java world as well as the .NET world with one XML-based language. Every instruction understood by the virtual machine, there is one XML-tag that represents this instruction. XMLVM is based on a fine-granular syntax which means that the complete syntax of XMLVM is accessible to an XML-parser.

To facilitate the generation of XMLVM, we have implemented various translators. The idea is to hide the complexities of XMLVM from the programmer who will only "see" his or her high-level programming language. We have written two translators: the first one converts a Java class file to XMLVM and the second one converts a .NET Intermediate Language program to XMLVM (see Figure 1).

XMLVM programs can readily be mapped to other high-level programming languages. This translation can easily be done by an XSL-stylesheet that maps XMLVM-instructions one-to-one to the target language. Since XMLVM is based on a simple stack-based machine, we simply mimic a stack-machine in the target language. For those high-level languages that do not support a goto-statement (such as JavaScript), we can remove those goto-statements and replace them with a combination of a loop- and multi-level exit-statements.

Applications

Middleware defines a software layer between the network and the application to facilitate the development of distributed applications. Any middleware technology supports different programming languages. The way XMLVM is used here

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 20–21, 2005.
© Springer-Verlag Berlin Heidelberg 2005

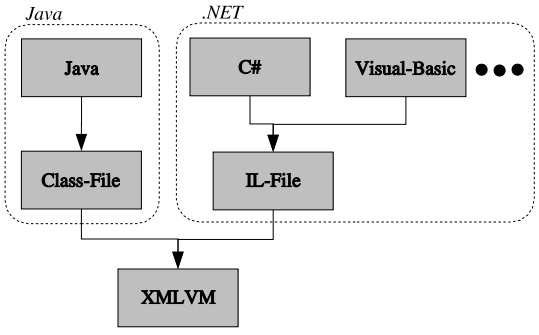


Fig. 1. Generating XMLVM

is to translate functional tests written in Java to XMLVM and then mapping XMLVM to C++. Two XMLVM transformations – both written as XSL-stylesheets – are applied to change the API. The first transformation adapts the CORBA-API. The second transformation does the same except for the JUnit API. The resulting XMLVM program is then translated to C++ by a third stylesheet.

The transformation chain explained above works well for many functional tests that ordinarily would have to be written for all languages supported by CORBA. The resulting C++ functional tests are not efficient from a runtime perspective (due to the overhead of mimicking a stack-machine), but performance is not important for functional testing. Further details can be found in [2].

We have used XMLVM in another project called XML11, which is an abstract windowing protocol inspired by the X11 protocol developed by MIT. In order to reduce latencies for interactive applications, XML11 also includes a code migration framework that allows to migrate part of the business logic to the end-device. We use XMLVM to translate the business logic to a language that is supported by the end-device, which is done by applying an appropriate XSL-stylesheet to the XMLVM program. E.g., for web browsers the business logic is translated to JavaScript which is universally supported by all major browsers including Microsoft's Internet Explorer. Further details can be found in [1].

References

1. Arno Puder. XML11 - An Abstract Windowing Protocol. *PPPJ*, 2005.
2. Arno Puder and Limei Wang. Cross-Language Functional Testing for Middleware. *In TestCom*, LNCS, Montreal, 2005. Springer.

Software Design of Electronic Interlocking System Based on Real-Time Object-Oriented Modeling Technique

Jong-Sun Kim[1], Ji-Yoon Yoo[1], and Hack-Youp Noh[2]

[1] Department of Electrical Engineering, Korea University,
1, 5-ga, Anam-dong, Seongbuk-gu, Seoul, Korea
{kids, jyyoo} @korea.ac.kr
[2] Korean Agency for Technology and Standards, Ministry of Commerce,
Industry & Energy, 2, Joongang-dong, Gwacheon-Si, Gyunggi-do,
Seoul 427-716 Korea
noh9289@ats.go.kr

1 Introduction

Electronic interlocking systems using micro-computer are developed to overcome the problems of conventional relay interlocking systems and to minimize the cost and the maintenance requirements when the system needs to be rebuilt or expanded.[1] However, it is very difficult to diagnose the root cause in case of a single device problem since there are multiple causes. Therefore, guaranteeing the stability to a equivalent level to relay interlocking systems is the main requirement for electronic interlocking systems to be benefit. This can be accomplished by a careful design of both the hardware and software and their interface. The stability of the interlocking software is determined by not only its reliability and efficiency of interlocking implementation but also by the convenience of maintenance. The method of real-time system development and (the) method of conventional data have an error of the confidential side, error detecting and error recovery, problem of exception situation processing, reusability of software of process and maintenance aspect and so on. The method for supplement shortcoming of these methods is real-time software development methodologies of object center. These methodologies play important role in solving complexity system, maintaining and requiring increased problems of software quantity as applying to object intention concept. But, because these methods are putting emphasis on analysis than design method, which are quitting emphasis on object structure among the analyses, development of real-time software has lacking aspects. A design approach for developing interlocking software to improve the problems of existing systems was proposed in this paper. A design and modeling strategy based on the Real-time Object-Oriented Modeling (ROOM)[2] procedure, which is the most appropriate approach in the initial stage of real-time software development, is proposed. Although it is an object-oriented method, it is a top-down design method that is similar to the structural analysis method based on the ROOM that is effective for real-time problems; therefore, it is not only convenient for standardization, expansion, and maintenance but also can contribute to improved reliability and stability of the electronic interlocking system.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 22 – 23, 2005.
© Springer-Verlag Berlin Heidelberg 2005

2 The Software Design Strategy of Interlocking System

The design method proposed in this paper is based on the ROOM method for building a precise and simple system model and for designing a recursive processor structure. The interlocking system must respond to the incoming response and process the acquired data within the limited time. The system is complicated by the fact that it must be operated on real-time. All system device components are viewed by objects and its modeling repetitively to solve the complexity of the system.

The design strategy is based on a Modeling Heuristic search Technique (MHT), which recognizes the specific requested condition and performs the detail designs based on the requested condition of system. A Message Sequence Chart (MSC) is formulated after analyzing the required scenario to model the internal system structure. The objects recognized during the internal structure modeling are modeled to determine their logical relations. After such system modeling, the optimized model is created by the required scenario. A gradual and repetitive approach for modeling are created during this process, and each modeling period processes the increments of requested conditions. Repetition occurs when the classes created from the previous modeling period is re-examined. The modeling strategy is to 1) utilize the advantage of the new paradigm (object orientation), 2) include the powerful real-time concept, and 3) make it easy to build a precise and simple system model. In addition, it is possible to recognize the elucidative system structure and records and the requirements and design flaws can be detected in an early stage since an active model is provided by surmising all concepts of leveling.

3 Conclusion

A new reliable on-line interlocking handle control algorithm providing stability as well as standardization, expansion ability, and convenience of maintenance has been proposed. The new design strategy was designed so that it provides to provide a reliable control system through repetitive process modeling. Another design criterion was to be able to verify the control system requirements by modeling systems during a short period, which enabled detection of design flaws and thus enhanced the precision.

The new method designs the control algorithm as a module for each unit, and the complex data structure of the interlocking data was easily recognizable since it was designed as a file structure that displays interlocking conditions similar to the connection status of a railway line.

References

1. A.H. Cribbens, : Solid-State Interlocking (SSI) : An Integrated Electronic Signaling System For Mainline Railway. IEE Proc. Vol. 134. MAY (1987) 148 – 158
2. B. Selic, G. Gullekson, and P.T.Ward. : Real-Time Object-Oriented Modeling. John Wiley & Sons. (1994)

Ontology Based Negotiation Case Search System for the Resolution of Exceptions in Collaborative Production Planning

Chang Ouk Kim[1], Young Ho Cho[1], Jung Uk Yoon[2], Choon Jong Kwak[1], and Yoon Ho Seo[3,*]

[1] Dept. of Information and Industrial Systems Engineering, Yonsei University, Shinchondong 134, Seoul, Republic of Korea
{kimco, yhcho94, cjkwak}@yonsei.ac.kr
[2] Mobile Handset R&D center, LG Electronics Inc., Seoul, Republic of Korea
[3] Dept. of Industrial Systems and Information Engineering, Korea University, Anamdong, Seoul, Republic of Korea
yoonhoseo@korea.ac.kr

Abstract. In this paper, we present an ontology based negotiation case search system that supports contractors to solve exceptions generated during the operation of supply chain.

1 Introduction

In the real world, many exceptional problems that violate the conditions specified in the original contracts occur, and the companies should resolve the problems through negotiation with additional time and cost. In general, the negotiation results for exception handling can be classified into three types: (1) finding the substitute of product, (2) finding the alternative of seller, (3) finding resolved past negotiation cases helpful for resolving current exception.

In this paper, we present an ontology based negotiation case search system that supports contractors to resolve exceptions generated during the operation of supply chain. The system intelligently searches past negotiation cases that seem to be useful for resolving current exception. The search results are of the three types mentioned above. Technically, we created an OWL (Ontology Web Language) based contract ontology by referring to an internationally standard contract document (see Figure 1). Also, we propose a negotiation case template (structure) which consists of original contract, exception contract, and resolved contract, in order to search past negotiation cases using the inference functions provided by description logic: (1) original contract contains information about the contract content being in conflict situation, (2) exception contract contains information about the exception of the original contract and additional exceptions generated due to the exception, (3) resolved contract contains information about the resolution of the exceptions in the exception contract. Consequently, the past negotiation cases are stored in case repository in our system by the negotiation case template, and contractors can search easily the past negotiation cases to resolve the exceptions of the original contract using our system.

* Corresponding author.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 24 – 25, 2005.
© Springer-Verlag Berlin Heidelberg 2005

2 OWL Based Contract Ontology

In Figure 1, Contract has three classes of Participant, Product, and Contract Condition as its properties. Also, each class has a special property to enable inference and inherits it to their subclasses. Detail explanation is as follows.

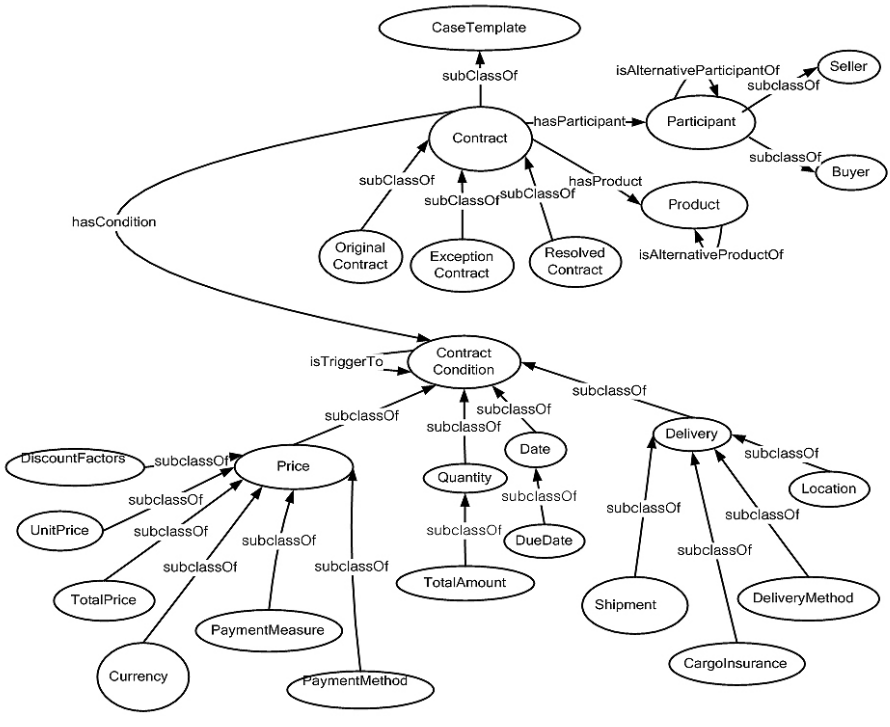


Fig. 1. Contract Ontology

- Participant has Seller and Buyer as its subclass and inherits its symmetric property isAlternativeParticipantOf. A buyer can obtain the information about sellers that manufacture the same product through this property, and a seller can also obtain the information about buyers that need the same product.
- Product has the symmetric property isAlternativeProductOf. A buyer can find the alternatives of the contracted product through this property. This property thus enables the buyer to solve exceptional problem by changing product.
- ContractCondition has Price, Delivery, Date, and Quantity as its subclasses and each subclass has some subclasses again as shown in Figure 1. ContractCondition inherits its transitive property isTriggerTo to all the subclasses. With this property, contractors can find which other exceptions simultaneously occur or are triggered after with the exception being considered.

These three properties are the core of inference used for handling exceptions in our system.

Enhanced Workflow Models as a Tool for Judicial Practitioners

Jörn Freiheit[1], Susanne Münch[2], Hendrik Schöttle[2], Grozdana Sijanski[2], and Fabrice Zangl[3]

[1] Max-Planck-Institute for Informatics
freiheit@mpi-sb.mpg.de
[2] Institute for Law and Informatics, Saarland University
{s.muench, h.schoettle, g.sijanski}@mx.uni-saarland.de
[3] Institute for Information Systems, DFKI GmbH,
66123 Saarbrücken, Germany
zangl@iwi.uni-sb.de

Abstract. In the past, attempts were made to make law and justice more accessible to general audience and to legal practitioners using models of legal texts. We present a new approach to make the judicial workflows easier to understand. By using process modelling methods, the developed representation emphasises on improving transparency, on promoting mutual trust and on formalising models for verification. To design semi-formal models interviews are used as well as legal texts are consulted. These models are formalised in a second step. The models are enhanced with hierarchies, modules and the generation of different views. Language problems are also treated. The subsequent formalised models are used to verify trigger events and timing of judicial workflows, which have very specific requirements in terms of periods of time and fixed dates. A new tool, *Lexecute*, is presented which gives new perspectives into justice and reveal new potentials for modelling methods in the field of justice.

The results presented in this paper have been achieved in the context of *eJustice*, an EU-project within the 6th framework programme (see www.ejustice.eu.com). eJustice is bringing together experts from the field of informatics and law, developing solutions to enable the European justice for a closer collaboration. One major part of eJustice, that will be presented here, is the development of a proper representation for judicial workflows.

We present the new tool *Lexecute* that combines a graphical representation of legal processes with a detailed description of all steps of these processes. Lexecute supports the work of judicial practioners for several reasons:

- A workflow model helps to visualise the process described. One can get a quick overview over the procedure without having to concern huge texts. A visualisation can be realised in such a self-descriptive way that it is understandable even by amateurs and not only by legal experts.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 26–27, 2005.
© Springer-Verlag Berlin Heidelberg 2005

- Visualisation promotes mutual trust. If the applicant knows where and how his request is processed and whom he can turn to in case of questions he brings enhanced trust into the process.
- The visualisation of processes in a modelling language allows formalisation. Formalisation allows verification. The next step can be automation then, or at least support of the processing of judicial workflows by computer technology.

Legal sources are the fundaments of judicial decisions and define the legal processes. Although some of them are meant for the organisation of work, there is hardly a process described by one single legal source only. Hence, the representation of judicial processes in workflow models can be of great support, just as it is in the private sector. In the service sector, process models are even used as an engineering method for services and could correspondingly be used for the development of new laws in the future.

We call our representation *enhanced workflow models* since they contain more information than a typical business workflow model. The information needed by judicial practitioners is too complex to be displayed within the graphical representation of a workflow model. Thus, we decided to enrich the workflow model with additional properties that are not shown in the graphical representation but in a separate *info-box*. By clicking on an element in the workflow model, the info-box returns information that would otherwise overcrowd the workflow model. The info-box is generated automatically. This box is the most important interface to the user. It contains information on the legal basis of an element, of documents, short descriptions, navigational information etc.

Two methodical mechanisms have been developed in main workflow modelling methods and are implemented in Lexecute: hierarchy and modularisation.

A hierarchy allows to refine the workflows and their functions stepwise and is represented by a tree structure. A Workflow is composed of several functions executed in a time-logical sequence and each function is supported by a workflow (except for the lowest function in the hierarchy).

The second possibility to reduce the complexity of a workflow model is the modularisation. Modules are self-contained parts of a workflow that have a defined in- and output and that can have multiple usage. They can be handled more flexibly either by representing them in a strictly logical sequence (without a time sequence) or by defining a set of modules for a specific domain or usage. The advantages of modularisation are that they are reusable (in this case it is sufficient to model and view a module only once, even if it is used several times in a workflow) and that they are exchangeable, e.g. the module `serve a claim by mail` can be replaced by the module `serve a document electronically`.

When modelling trans-national workflows, the language barrier forms an obstacle and complicates the understanding. However, the judicial legal terminology differs not only from language to language but also from country to country. In our models the original terms are used and translations are presented by moving the mouse over the terms depending on the country in which our tool is used.

Semantics of Information Systems Outsourcing

H. Balsters and G.B. Huitema

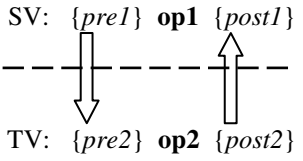
University of Groningen, Faculty of Management and Organization, The Netherlands
{h.balsters, g.b.huitema}@rug.nl

Research Overview and Results

Businesses are in nature dynamic and change continuously. Because of different economic prospects they grow in size and portfolio or just the other way they have to reduce one of these aspects. There are several ways to accomplish growth or reduction. A smooth way may consist of outsourcing parts of one's non-core business processes to specialized parties in the market. A variety of outsourcing models have been developed ([6]). Outsourcing can range from having all the business process (such as development, maintenance and operations) performed by an outsourcing partner, up to having a contract with a partner performing only one single business task. In our work we concentrate on conceptual modeling of outsourcing information systems, where outsourcing in the context of information systems will be defined as delegating a part of the functionality of the original system to an existing outside party (the supplier). Such functionality typically involves one or more operations (or services), where each operation satisfies certain input- and output requirements. These requirements will be defined in terms of the ruling service level agreements (SLAs). We provide a formal means to ensure that the outsourcing relationship between outsourcing party and supplier, determined by a SLA, satisfies specific correctness criteria. These correctness criteria are defined in terms of consistency and completeness between the outsourced operation and the associated operation offered by the supplier. Our correctness criterion will concern mappings between an existing outsourcer schema and an existing supplier schema, and will address both semantical and ontological aspects pertaining to outsourcing. Formal specifications as offered in our work can prove their value in the setup and evaluation of outsourcing contracts. We will perform our analysis within the modeling framework based on the UML/OCL formalism ([8,9]). The Object Constraint Language OCL offers a textual means to enhance UML diagrams, offering formal precision in combination with high expressiveness. In [1] it has been demonstrated that OCL has at least the same expressive power as the relational algebra, (the theoretical core of the relational query language SQL), thus making OCL a very powerful language for specification of constraints, queries and views. Also, UML is the de facto standard language for analysis and design in object-oriented frameworks, and is being employed more and more for analysis and design of information systems, in particular information systems based on databases and their applications. We define so-called *exact views* [1,2,3] on top of existing information systems in order to eventually capture the formal requirements of the outsourcing relation. Exact views belong to the domain of data extraction and data reconciliation ([2,3,4,5,7]), and have the property that they are correctly updatable, in the sense that any update on an exact view corresponds to

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 28 – 29, 2005.
© Springer-Verlag Berlin Heidelberg 2005

exactly one combination of correct updates on the base classes it stems from ([3]). A SLA will be given precise specifications in terms of pre- and post-condition statements on operations in terms of OCL. In general, a targeted supplier has to abide to the following (abstract) outsourcing schema.



This schema (called an *ω-schema*) reads as follows: **op2** is a *correct outsourcement* of **op1**, if and only if pre-condition *pre1* logically implies pre-condition *pre2*, and post-condition *post2* logically implies post-condition *post1*. An ω-schema prescribes a consistency and completeness condition with respect to pre- and post conditions of the outsourcer- and the supplier operations involved. The challenge of finding a correctly implemented outsourcing now boils down to constructing a mapping from a view SV (on the source model SM) to a view TV (on the target model TM), such that this mapping respects an ω-schema for outsourcing as described above. Our work translates recent results from the field of data integration [2,3,4,5,7] to the field of outsourcing, typically employing the novel concept of exact view.

References

1. Balsters, H. ; Modeling Database Views with Derived Classes in the UML/OCL framework;«UML» 2003 6th Int. Conf.; LNCS 2863, Springer, 2003
2. Balsters, H., de Brock, E.O.; An object-oriented framework for reconciliation and extraction in heterogeneous data federations; Proc. 3rd Int. Conf. Advances in Information Systems, LNCS 3261, Springer, 2004
3. Balsters, H., de Brock, E.O.; Integration of integrity constraints in federated schemata based on tight constraining; Proc. OTM Confederated International Conferences CoopIS, DOA, and ODBASE , LNCS 3290, Springer, 2004
4. Bouzeghoub,M., Lenzerini, M; Introduction to: data extraction, cleaning, and reconciliation, Special issue; Information Systems 26 ; Elsevier Science, 2001
5. Lenzerini, M.; Data integration: a theoretical perspective; ACM PODS'02, ACM Press 2002
6. Looff, L. A. de ; Information systems outsourcing decision making: a framework, organizational theories and case studies, Journal of Information Technology, Vol. 10 Issue 4, p281 - 297, 1995.
7. Miller, R.J., Haas, L.M., Hernandez, M.A.; Schema mapping as query discovery; Proc. 26th VLDB Conf.; Morgan Kaufmann, 2000
8. Response to the UML 2.0 OCL RfP, Revised Submission, Version 1.6,January 6, 2003
9. Warmer, J.B., Kleppe, A.G.; The object constraint language; Addison Wesley, 2003

Integration of Heterogeneous Knowledge Sources in the CALO Query Manager*

José Luis Ambite[1], Vinay K. Chaudhri[2], Richard Fikes[3], Jessica Jenkins[3], Sunil Mishra[2], Maria Muslea[1], Tomas Uribe[2], and Guizhen Yang[2]

[1] USC Information Sciences Institue, Marina del Rey, CA 90292, USA
[2] Artificial Intelligence Center, SRI International, Menlo Park, CA 94087, USA
[3] Knowledge Systems Laboratory, Stanford University, Stanford, CA 94305, USA

1 Introduction

We report on our effort to build a real system for integrating heterogeneous knowledge sources with different query answering and reasoning capabilities. We are conducting this work in the context of CALO (Cognitive Assistant that Learns and Organizes), a multidisciplinary project funded by DARPA to create cognitive software systems.

The current project is targeted at developing personalized cognitive assistants (CALOs) in an office environment where knowledge about emails, schedules, people, contact information, and so on is distributed among multiple knowledge sources. A CALO must be able to access and reason with this distributed knowledge. We have encapsulated this functionality in a CALO module called Query Manager. Two typical example queries that need to be answered by Query Manager are:

1. *Which meetings will have a conflict if a particular meeting runs overtime by an hour?* Answering this query requires retrieving the ending time of the meeting from IRIS (a personal information knowledge source), computing the new ending time using Time Reasoner (a special reasoner), and querying PTIME (a personal time management system using a constraint solver as its main reasoning engine) with relevant information.
2. *Who was present in the meeting in conference room EJ228 at 10 a.m. this morning?* Answering this query requires retrieving knowledge (expressed as rules) about how to identify meeting participants from a knowledge base

* This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA, or the Department of Interior-National Business Center (DOI-NBC).

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 30–32, 2005.
© Springer-Verlag Berlin Heidelberg 2005

called KM, and doing inference based on these rules using another independent knowledge base called MOKB (which stores analysis results on images captured by meeting room cameras).

The examples above illustrate the challenges facing Query Manager in integrating knowledge sources and answering queries of different sorts. However, users of Query Manager do not have to worry about how knowledge is expressed (e.g., as ground facts or axioms) or distributed in the system. Queries only need to be formulated using CALO Ontology. Query Manager will automatically determine which knowledge sources are needed and how to produce the answers.

2 The CALO Query Manager

The architecture of the CALO Query Manager is depicted in Figure 1. Reasoners in Query Manager are organized in a hierarchical fashion. The Query Manager design is based on an object-oriented modular architecture for hybrid reasoning, called the JTP architecture. The main advantage of our design is that it supports encapsulation of reasoners and reasoner functionalities, and easy, incremental implementation and integration of reasoners and reasoning systems.

The entry point of Query Manager is Asking Control Reasoner. It uses two reasoning methods, iterative deepening and model elimination, to control the overall execution of a query. Asking Control Reasoner sends queries to Asking Control Dispatcher, which calls three reasoners in sequence: Rule Expansion Reasoner, Query Planner, and Assigned Goal Dispatcher. Rule Expansion Reasoner applies rules to the input query and produces expanded forms of a query. Query Planner produces an evaluation plan that groups and orders sets of literals in the query according to the capabilities of knowledge sources. Assigned Goal Dispatcher executes a query by dispatching (groups of) subgoals to different reasoners and knowledge sources per the query plan produced by Query Planner. Each knowledge source is integrated into Query Manager by implementing an asking reasoner per the JTP reasoner interface specification. These reasoners encapsulate the source-specific details of evaluating queries.

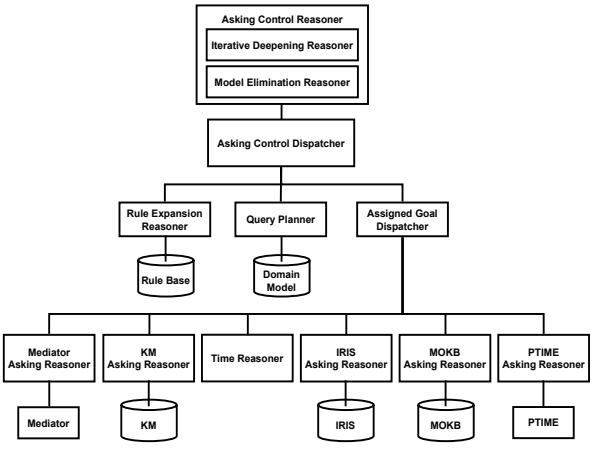


Fig. 1. The CALO Query Manager Architecture

3 Implementation

Query Manager has been implemented and deployed in a functioning CALO system. It plays a central role of answering queries to heterogeneous knowledge sources. The salient features of our system include the following: (1) Knowledge sources integrated into our system may return new subgoals as well as ground facts in response to queries; (2) The limited reasoning capabilities of individual knowledge sources are augmented by using a powerful query planner; (3) The reusable, object-oriented CALO Ontology provides a mediated schema that serves as the basis for integration; (4) Experimental results showed that our query planner runs efficiently in practice.

More details about our work on the CALO Query Manager are available in our online technical report at `http://www.ai.sri.com/pubs/files/1180.pdf`

Context Knowledge Discovery in Ubiquitous Computing

Kim Anh Pham Ngoc, Young-Koo Lee, and Sung-Young Lee

Department of Computer Engineering,
Kyung Hee University, Korea
{Anhpnk, sylee}@oslab.khu.ac.kr,
yklee@khu.ac.kr

Abstract. This article introduces the concept of context knowledge discovery process, and presents a middleware architecture which eases the task of ubiquitous computing developers, while supporting data mining and machine learning techniques.

1 Introduction

Many current Ubiquitous systems, such as Gaia [1], are using reasoning engines to infer high-level information from the low-level context data. However, the performance is limited due to the complications in composing rules for the rule-based reasoners, or calculating the uncertainty in probabilistic reasoners. In this paper we introduce context knowledge discovery process (CKDD) which was realized in CAMUS - our middleware architecture for context-aware systems. We also illustrate CKDD by explain the rule learning mechanism of CAMUS.

2 Context Knowledge Discovery in CAMUS Middleware

Context knowledge discovery (CKDD) differs from original knowledge discovery (KDD) in several aspects. While KDD works with transactional data in business and commercial systems, CKDD deals with context data in context-aware systems. While KDD normally discovers the interesting patterns of customers and sales, CKDD tries to model the users and their behavior, also tries to "understand" the needs of user so that a context-aware system can satisfy those needs in a ubiquitous manner.

CKDD is the core function of Learning and reasoning modules in CAMUS [2], a unified middleware framework for context-aware ubiquitous computing. The CKDD process includes 4 main steps: i) Context data preprocessing, which includes ontology mapping, context summary [3] or aggregation operations; ii) User identification (using RFID, user tag, badge, PDA, PC login...) and context recognition (using neural network, Bayesian network...); iii) Context data mining, which mines association rules, classification rule sets and clusters, to provide input to learning step; iv) Learning, which is illustrated by the rule learning algorithm and mechanism in Fig. 2.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 33 – 34, 2005.
© Springer-Verlag Berlin Heidelberg 2005

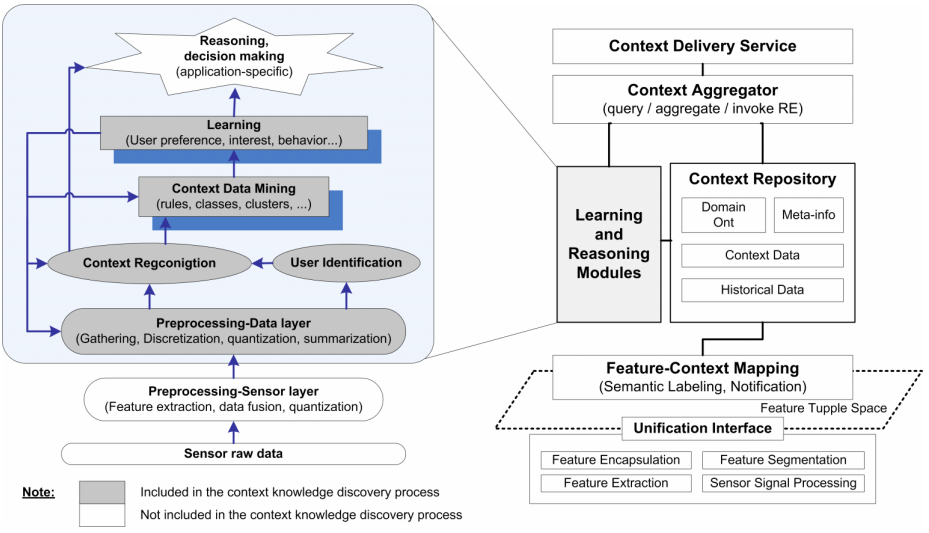


Fig. 1. CKDD, its position in the ubiquitous data inquiry and knowledge management process, and its correlation to CAMUS architecture

i/ *Learning the best rule*, each rule is assigned a utility function.

ii/ *Removal of covered example.s*

iii) Iterations: Repeat step i) and ii) until the number of attributes in a frequent set, or the number of examples covered by a new rule reach a limit.

iv) *Iteratively updating the rule set*: update the Utility of each rule based on application's reponse, remove low-utility rules and learn new rules from the updated training data.

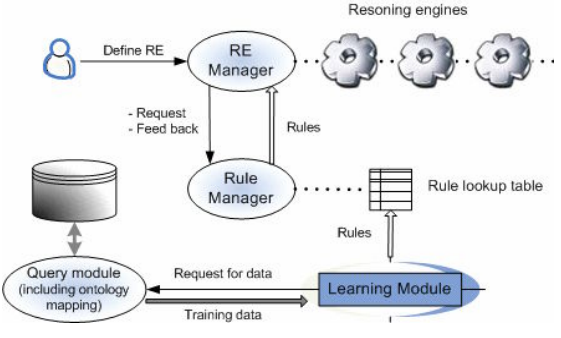


Fig. 2. Rule learning algorithm and mechanism in CAMUS

References

[1] Anand Ranganathan, Roy H. Campbell: A Middleware for Context -Aware Agents in Ubiquitous Computing Environments. In ACM/IFIP/USENIX International Middleware Conference, Brazil, June, 2003.

[2] Anjum Shehzad et. al. A Comprehensive Middleware Architecture for Context-Aware Ubiquitous Computing Systems. Accepted for publication, ICIS 05.

[3] Faraz Rasheed, Yong-Koo Lee, Sungyoung Lee, "Context Summarization and Garbage Collecting Context", Ubiquitous Web Systems and Intelligence 2005.

Ontology-Based Integration for Relational Data

Dejing Dou and Paea LePendu

Computer and Information Science, University of Oregon
{dou, paea}@cs.uoregon.edu

Motivation. Recent years gave witness to significant progress in database integration including several commercial implementations. However, existing works make strong assumptions about mapping representations but are weak on formal semantics and reasoning. Current research and practical application calls for more formal approaches in managing semantic heterogeneity [3].

Framework and Results. In this paper, we briefly describe an ontology-based approach that uses a first order theorem-prover for information integration. We have also built OntoGrate to evaluate our approach and describe its architecture here. Throughout, we refer to the term *ontology* as *the formal specifications of the vocabularies of concepts and the relationships among them.* Also, for us, integration has two aspects: (i) query answering and (ii) data translation.

Mappings between schemas are essential to integration and require an adequate representational language. SQL views have been widely used, but mappings can also be represented with other languages having formal semantics (e.g., Datalog, XQuery). Instead, we choose a more expressive first order ontology language, Web-PDDL, to represent complex mappings between schemas as *bridging axioms* (first order mapping rules). The advantage in doing so is the specialized theorem prover, OntoEngine [2], can then perform query answering and data translation while formally preserving semantics. We refer to this process as *inferential data integration* because it uses sound inference by either forward chaining or backward chaining.

Two databases in the online sales domain, Stores7 from Informix and Nwind from Microsoft, serve as examples in Figure 1. First, we define a super ontology for SQL to express concepts such as aggregate functions and integrity constraints that exploit desirable features of database systems via ontology inheritance. Then, by a simple process, we translate each database schema into its own ontology. Next, we define mappings between each ontology using bridging axioms and call this our *merged ontology.* Finally, syntax translators that we developed allow OntoEngine to access actual relational data by transforming atomic queries from Web-PDDL to SQL. These elements are summarized in the architecture of OntoGrate shown in Figure 1. Therefore, the user can submit a query or translation request which OntoEngine fulfills by either backward or forward chaining on the bridging axioms in the merged ontology using actual data retrieved by the SQL syntax translators.

Although our Web-PDDL-to-SQL translators only handle atomic queries for now, the inference mechanisms can still process more complex conjunctive

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 35–36, 2005.
© Springer-Verlag Berlin Heidelberg 2005

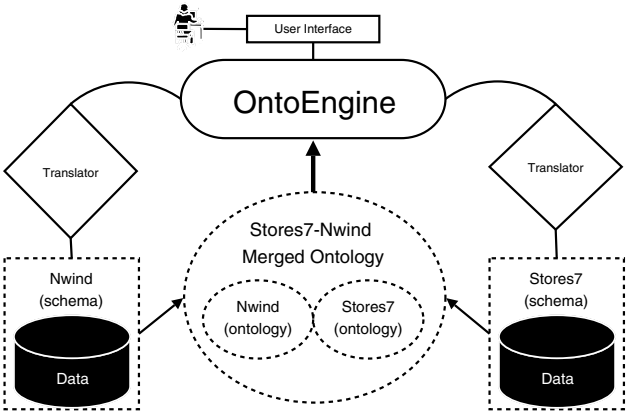


Fig. 1. Architecture for OntoGrate. The system integrates two sales databases using OntoEngine, query syntax translators and a merged ontology.

queries in a way similar to query decomposition. However, conjunctive query translators might take direct advantage of efficient database join optimizers automatically.

Preliminary tests of OntoGrate show linear performance based on the number of answers (for querying) or facts (for translation). Over 25,000 records can be processed per minute for query answering and 10,000 per minute for data translation on an unremarkable personal laptop computer. New data structures in our forward chaining algorithm should further improve translation performance.

Conclusion and Future Work. In conclusion, we have developed an ontology-based approach to integrate heterogenous relational databases using *inferential data integration* that exploits both the expressivity of first order logic and the desirable features of SQL by using ontology inheritance. Preliminary tests of OntoGrate are promising for relational database integration. Immediate future work will include conjunctive query reformulation and efficient data structures for OntoEngine. In the long term, we anticipate that logical approaches will prove not only instrumental in integrating relational databases but also other structured data such as those in the Semantic Web [1].

References

1. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.
2. D. Dou, D. V. McDermott, and P. Qi. Ontology Translation on the Semantic Web. *Journal of Data Semantics*, 2:35–57, 2005.
3. A. Y. Halevy, N. Ashish, D. Bitton, M. J. Carey, D. Draper, J. Pollock, A. Rosenthal, and V. Sikka. Enterprise Information Integration: Successes, Challenges and Controversies. In *Proceedings of SIGMOD*, pages 778–787, 2005.

AWeSOMe 2005 PC Co-chairs' Message

We wish to extend a warm welcome to AWeSOMe'05, The First International Workshop on Agents, Web Services and Ontologies Merging. This workshop will be held in conjunction with the On The Move Federated Conferences and Workshops 2005 (OTM'05).

The current and future software needs are towards the development of large and complex Intelligent Networked Information Systems, covering a wide range of issues as required for the deployment of Internet- and Intranet-based systems in organizations and for e-business. The OTM'05 Federated Conferences and Workshops provide an opportunity for researchers and practitioners to their background to different emerging areas such as Data and Web Semantics, Distributed Objects, Web Services, Databases, Workflow, Cooperation, Interoperability and Mobility.

Web services are a rapidly expanding approach to building distributed software systems across networks such as the Internet. A Web service is an operation typically addressed via a URI, declaratively described using widely accepted standards, and accessed via platform-independent XML-based messages.

Emerging ontologies are being used to construct semantically rich service descriptions. Techniques for planning, composing, editing, reasoning and analyzing about these descriptions are being investigated and deployed to resolve semantic interoperability between services within scalable, open environments.

Agents and multi-agent systems can benefit from this combination, and can be used for web service discovery, use and composition. In addition, web services and multi-agent systems bear certain similarities, such as a component-like behavior, that can help to make their development much easier.

The set of technical works presented here focuses on topics that span the interfaces between intelligent agents' technology, the use ontologies for semantic normalization, and infrastructure for web services. These papers concentrate on some key research issues (such as: service discovery, composition, and delegation; service security and policy definition; and architectures for semantic services and agents) pointing out some important problems that would need to be addressed in order to realize the promise of semantically-aware Service-Oriented Computing.

This workshop could not have taken place without considerable enthusiasm, support and encouragement as well as sheer hard work. Many people have earned the thanks of those who attended and organized AWeSOMe'05. In particular, we would like to thank:

- The many supporters of OTM'05 for their contributions to the conference. Many of these people have been involved with the OTM conferences for several years.
- The members of the AWeSOMe'05 Program Committee who gave their time and energy to ensure that the conference maintained its high technical quality and ran smoothly. The many individuals we owe our thanks to are listed in this volume.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 37–38, 2005.
© Springer-Verlag Berlin Heidelberg 2005

– All those who submitted to the workshop. The standard set was higher than our expectations and reflected well on the research work in the community.

We would also like to acknowledge the organizers of the OTM'05 conferences for the support and encouragement they extend to this workshop. The close cooperation between AWeSOMe'05 and the OTM'05 organization allows us to contribute to the growth of this research community.

August 2005

Pilar Herrero, Universidad Politécnica de Madrid
Gonzalo Méndez, Universidad Complutense de Madrid
Lawrence Cavedon, Stanford University
David Martin, SRI International
(AWeSOMe'05 Program Committee Co-Chairs)

Document Flow Model: A Formal Notation for Modelling Asynchronous Web Services Composition

Jingtao Yang, Corina Cîrstea, and Peter Henderson

School of Electronics and Computer Science, University of Southampton,
Southampton SO17 1BJ, UK
{jy02r, cc2, ph}@ecs.soton.ac.uk

Abstract. This paper presents a formal notation for modelling asynchronous web services composition, using context and coordination mechanisms. Our notation specifies the messages that can be handled by different web services, and describes a system of inter-related web services as the flow of documents between them. The notation allows the typical web services composition pattern, asynchronous messaging, and has the capability to deal with long-running service-to-service interactions and dynamic configuration behaviors.

1 Motivation

A service-oriented application is a composition of web services aimed to achieve certain business goals [1,2]. The goals are fulfilled by service-to-service interactions. Our work intends to capture the behavior of service level interactions in a formal notation for specifying web services composition. The language is general and can be applied to various business environments, in order to support web services composition, automation and validation.

First of all, a service-to-service interaction is not just a transaction. Web service transactions are a subset of service interactions [3]. A transaction is a group of logical operations that must all succeed or fail as a group. Currently most service composition specifications such as BEPL4WS only provide mechanisms to support long-running transactions by providing two-phase commit protocols and compensation activities [4]. However, an interaction may include multiple transactions, and can last longer than a transaction. For example, in a banking application, customers pay an estimate amount of money for their utility expense in advance. When the actual numbers arrive a month later, perhaps longer, the difference has to be paid to the customer's account or utility provider's bank account. This interaction is completed by two payment transactions. As long-running interactions are the basis of modern enterprise applications, they should also be considered when specifying services composition.

We believe that service composition should be more dynamic. Storing interaction states in short-lived instances at web containers, as BPEL4WS does, means that services can not be replaced in the middle of interactions. But to meet dynamically-changing business, web service applications are often required to be recomposable.

Context management is a more fundamental requirement than transactions in some business environments [3]. A context allows web services to share information such as

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 39 – 48, 2005.
© Springer-Verlag Berlin Heidelberg 2005

message correlations, security tokens and so on. We use the context mechanism to model service interactions by giving each interaction a context. Using this context, a service can continue an interaction that was previously operated by another service. Our model allows interactions and contexts to be structured hierarchically. Thus an interaction could be coordinated by not only a certain service, but also distributed services.

We model services composition by describing the messages exchanged between web services. Each service specifies its contributions to an interaction by updating the interaction state or coordinating its context.

In a previous paper [5], we investigated the requirements for dynamic configurations for service-oriented systems, and argued that long-running interactions are necessarily asynchronous. A notation for describing such systems, called the Document Flow Model (DFM), was also introduced in loc. cit. In this paper, we give a complete formal syntax, and an informal semantics for this notation. We also discuss the use of a special coordination service, in conjunction with a global store, as a means to coordinate interactions over dynamical web services. A formal operational semantics for our notation has also been developed, and is described in [6]. The present paper complements this work by focusing on the use of context and coordination mechanisms in modelling complex web service interactions.

The paper is structured as follows: Section 2 gives the formal syntax, and an informal semantics for our notation. Section 3 uses a job submission example to illustrate the use of DFM in specifying web services composition, and to discuss the use of a special coordination service. Section 4 further discusses the capability of dealing with dynamic configurations, while Section 5 summarizes our approach.

2 Document Flow Model

Our notation describes a system of web services as a set of messages that can be sent from individual services, and the consequences for other services of receiving them. Because messages are basically XML documents, we call it Document Flow Model. A hierarchical (tree) data structure called a document record is used in DFM to abstractly model systems that are eventually realized using XML documents.

The DFM notation is intended to model systems composed by sets of independent web services, orchestrated by asynchronous messages. Since we are not interested in the functionality and performance for each service, we model a web service as a collection of outgoing messages sent in response to an incoming message. Two kinds of communication are supported: one-way communication, which amounts to a service receiving a message, and notification, which amounts to a service sending a message. Request-response and solicit-response [7] conversations are modeled as a one-way communication plus a notification communication.

DFM provides support for long-running interactions and dynamic configurations. One aspect of the ability to simply unplug something in the middle of an interaction and plug in a substitute, is whether or not the component has state. Replacing a stateful component with another is always more difficult than replacing a stateless component with another [8, 9]. This is one of the reasons why one of the main design criteria for web services is that they should be stateless [1]. Our notation models an interaction via stateful messages passed around stateless web services. A context is given to each message to identify the interaction it belongs to. A decentralized context

propagation mechanism is used to structure interaction-related data. A persistent component, a ContextStore, is used to maintain the execution state. A stateless web service simply reacts to incoming messages, and updates the state within the ContextStore when necessary. By coordinating all state with the persistent component, an interaction can carry on even if the system configuration has changed.

The Basic Specification Structure. A DFM specification is built from message definitions, or messagedefs. A web service is described by a collection of messagedefs, specifying the messages which the web service receives and operates on.

messagedefs ::= messagedef | messagedef messagedefs
messagedef ::= **OnMessage** message msgdefbody

A web service is accessed by XML messages. In DFM, each messagedef defines the service response to an incoming message: when an incoming message matches the message pattern in messagedef, the corresponding actions in msgdefbody are triggered.

The Message Definition Body. A message definition body, msgdefbody, defines the actions to be carried out when an incoming message matches a certain pattern. Possible actions include storing a document in a document store and sending a message.

msgdefbody ::= idaction storebody sendbody
storebody ::= _ | storeaction storebody
sendbody ::= _ | sendaction sendbody | csendaction sendbody

A msgdefbody may contain three pieces of information, idaction, storebody and sendbody, in this particular order; any of these can be absent. The idaction describes some new identities used to identify interactions started as a result of a message being acted upon. The storebody describes the set of store actions to be carried out, before the (possibly conditional) message sending actions described in sendbody are carried out in no particular order.

Actions. A message definition may contain essentially four kinks of actions: idaction, storeaction, sendaction and csendaction, as mentioned earlier.

idaction ::= _ | **generate new** ids
ids ::= id | id, ids

When a service starts a new business interaction, it usually creates a new identity to identify that interaction. An idaction specifies the identities generated in this way. The newly generated identities are universally unique, that is, identities generated by the same / different services are different; this can, for instance, be ensured by embedding information such as service identity, date, time and message content in each newly generated identity.

storeaction ::= **store** id**->**entry **in ContextStore**

A storeaction describes the action of storing a piece of information, an entry, into the ContextStore, under a particular identity id.

sendaction ::= **send** message
csendaction ::= **if** condition **then {** sendactions **}**
sendactions ::= sendaction | sendaction sendactions

A sendaction describes the action of sending out a message. A csendaction specifies one or more sendactions to be performed only when a certain condition (involving the current state of the ContextStore) holds. When the condition evaluates to true, the corresponding sendactions are taken. A simple control flow, a collection of non nested *if… then…*statements, is available in the DFM notation.

Conditions. A condition is a ContextStore evaluation expression, possibly containing logical operators. A simple condition evaluates to true when the specified entries are present in the ContextStore under the identity id, otherwise the condition evaluates to false. Conditions containing logical operators are evaluated in the standard way.

condition ::= **ContextStore** [id] **contains** entries
| condition **and** condition | condition **or** condition | **not** condition

XML Document Data Structure. Web Services interact with each other by messages which are essentially XML files. To model an XML message, we introduce a new data structure, a document record. A document record allows us to specify the properties of a document. A document record literal consists of a comma-separated list of colon-separated property name / value pairs, all enclosed within square brackets. In the document record, a property name is a string identifier, while a property value is an atom or another document record. A simpler form of document record contains no property names, only property values. In relation to XML, a document record is an XML element. We ignore XML attributes, important though they are in practice, because at the modelling level it is unnecessary to distinguish between nested attributes and nested elements. In a document record, an XML attribute is modeled by a property of that element.

A message is modeled in DFM as a document record with properties **to:**, **query:** and **function:**. The property values to and function are simple strings which describe the message receiver and the requested operation.

message ::= **[to:**to,**query:**query,**function:**function**]**

The property value query is a document record that refers to the message data, or message parameters.

query ::= element | **[from:**from,**query:**query,**context:**uid**]**
| **[from:**from,**query:**query,**result:**query,**context:**uid**]**

element ::= string | **[**elements**]**
elements ::= element | element, elements

Three types of queries are defined in DFM. The first one, element, is a simple document record with no property names, and the property value given by either a string or a list of elements. The second is a document record with **from:**, **query:** and **context:** properties. It includes the query initiator, content and identity. It is used, for example, when a web service initiates a business process by passing a query to other web services. The third is a document record with **from:**, **query:**, **result:** and **context:** properties. When a query has been completed, the results are put into a message together with the original query. As in the message document record, the **query:** and **result:** property values are further document records.

ContextStore. The systems modeled using DFM are concurrent: multiple interaction sessions are carried out at the same time. To maintain the system state, a unique identity is created and assigned to each interaction. The state is structured into document records, entries, and stored under the process identity in the ContextStore.

entries ::= entry | entry, entries
entry ::= **[from:**from,**query:**query] | **[from:**from,**query:**query,**result:**result]

An interaction is represented in the ContextStore by a set of entrys which point out that a query has been started, or that the query / its sub-queries have been completed.

3 An Example

We use a job submission system to illustrate our notation. When an application involves a large number of tasks, instead of buying a supercomputer, a more effective way is to deliver subtasks to different computers, and subsequently combine their results. Web services are one of the technologies used to implement such systems.

We have described that our notation allows an interaction to be coordinated by one service or by distributed services. In the following example, we use a Coordination Service to maintain the state of an interaction over stateless web services.
In a previous example [5], all the services participating in an interaction were able to access the state-maintaining component, ContextStore. In this example, the Coordination Service is the only service accessing the ContextStore. The reasons for this are as follows: First, restricting the access largely releases the concurrent control workloads on persistent components, especially in applications involving huge computing tasks. Second, by maintaining the state solely through the Coordination Service, this service can monitor the overall interaction, so that any failure can be detected and recovered timely. Finally, replacing a service with access to the state component is much more complicated than replacing a service with no access to it. Thus, the use of the Coordination Service makes our system more amenable to dynamic reconfiguration.

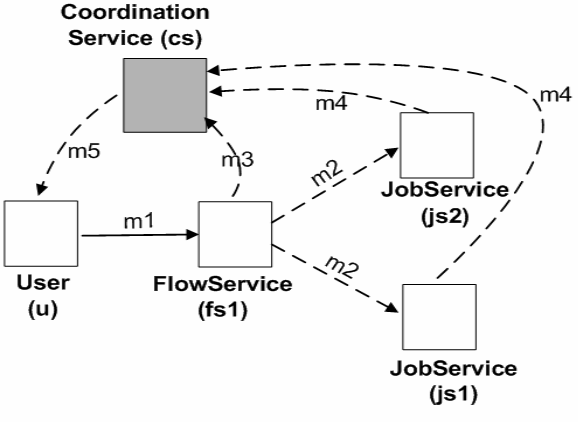


Fig. 1. A job submission system example

Our job submission specification defines three kinds of services. A FlowService, fs1, an orchestrating web service, makes use of JobServices. JobServices, js1 and js2, execute jobs and send results to the Coordination Service (m4). A Coordination Service, cs, coordinates job executions. In our example, when fs1 receives a message with a composed job (m1), it forwards two sub-jobs to js1 and js2 (m2). It also sends the job submission state to cs (m3). Once both sub-jobs have been completed, cs will return their combined result to the user who originally requested the job execution (m5). The different line formats in Fig. 1 are used to distinguish messages with different contexts.

```
OnMessage [ to:FS, query:[from:u,query:[ j1,j2],context:c], function:startjobs ]
    generate new uid
    send [ to:CS, query:[from:FS,query:[from:u,query:[ j1,j2],context:c],context:uid],
                                                                  function:jobssubmitted]
    send [ to:JS1, query:[from:FS,query:j1,context:uid], function:jobexecute ]
    send [ to:JS2, query:[from:FS,query:j2,context:uid], function:jobexecute ]
```

Fig. 2. A FlowService Specification

A FlowService (referred to by FS in our specification) requires a number of services, identified by JS1, JS2 and CS, to define its workflow. In the semantics of DFM, such service identifiers are mapped to actual services (e.g. js1, js2 and cs from Fig. 1), thus allowing a system to be dynamically-configured. In particular, the services corresponding to JS1 and JS2 could not only be JobServices, but also FlowServices with the extended capabilities of a JobService (see Section 4).

The user's query contains three parts: a simple query containing job tasks; the user who initiates it; and a context to identify the query. According to the specification in Fig. 2, when a FlowService receives a query from a user, it creates a unique identity, uid, to identify a new interaction. In this case, the job results will be delivered by CS. Thus, the FlowService informs CS that a new interaction has been started, by forwarding the user's query. FS only forwards the actual jobs to the two JobServices, thus preventing a direct interaction between users and JobServices. The FlowService sends out the messages concurrently, and then continues servicing other interleaved queries and replies.

```
OnMessage [ to:JS, query:[from:fs,query:job,context:uid], function:jobexecute]
    send [ to:CS, query:[from:JS,query:job,result:result,context:uid], function:jobcomplete]
```

Fig. 3. A JobService Specification

The JobServices execute jobs and forward their results together with the original queries and contexts, to CS. The original query is required to indicate to the CoordinationService which part of the interaction has been completed. When a

JobService (e.g. js2) receives a query containing two jobs (e.g. in the form [j2,j3]), it executes them and forwards the result [r2,r3] to CS.

```
OnMessage [ to:CS, query:[from:fs,query:[from:u,query:[ j1,j2],context:c],context:uid],
                    function:jobssubmitted ]
  store uid->[ from:fs, query:[from:u,query:[ j1,j2],context:c] ]  in ContextStore
  if  ContextStore[uid]  contains  [ from:fs, query:[ from:u,query:[j1,j2],context:c] ],
                                   [ from:js1, query:j1, result:r1 ], [ from:js2, query:j2, result:r2 ]
  then {send [ to:u, query:[from:CS,query:[from:u,query:[ j1,j2],context:c],result:[r1,r2],context:uid],
                    function:jobsreply ] }

OnMessage[ to:CS, query:[from:js,query:job,result:result,context:uid], function:jobcomplete]
  store uid->[from:js,query:job,result:result]  in ContextStore
  if  ContextStore[uid]  contains  [ from:fs, query:[ from:u,query:[ j1,j2], context:c] ],
                                   [ from:js1, query:j1, result:r1], [ from:js2, query:j2, result:r2 ]
  then {send [ to:u, query:[from:CS,query:[from:u,query:[ j1,j2],context:c],result:[r1,r2],context:uid],
                    function:jobsreply ] }
```

Fig. 4. A Coordination Service Specification

Because we assume that communications between services are asynchronous, the messages received by the Coordination Service are in an undetermined order. In our solution, each time the Coordination Service receives a message, it takes all contents except the context of the query and stores them into the ContextStore under the interaction's unique identity. The Coordination Service then checks if the ContextStore contains all the queries and results of that interaction. When sufficient information has been gathered, the Coordination Service replies to the user.

We can therefore see that the combination of a ContextStore and stateful interactions is sufficient to solve the problem of asynchronous coordinated interactions.

4 Discussion

The previous example shows how to describe asynchronous interactions in DFM. The DFM notation also aims to support dynamic configurations. Dynamic configuration is a very complex issue, especially in distributed systems. Our work is only concerned with high level interactions: we only model and analyse the integrity of an interaction, assuming that all the messages are safe and reliable.

To improve performance, some new services are added to the job submission system, as in Fig. 5. Specifically, the JobService js2 is replaced by a FlowService fs2 that has access to JobServices js2 and js3. The FlowService fs1 behaves as in the previous example, except that it will now send a job execution request (m6) to fs2 instead of js2. (The actual specification of the FlowService remains unchanged as far as receiving messages from users is concerned. The only change is in how the service identifier JS2 known to fs1 is mapped to an actual service.) Upon receiving a request from fs1, the FlowService fs2 passes two sub-jobs to the JobServices js2 and js3. This way, the jobs received by fs1 can be executed simultaneously by three JobServices.

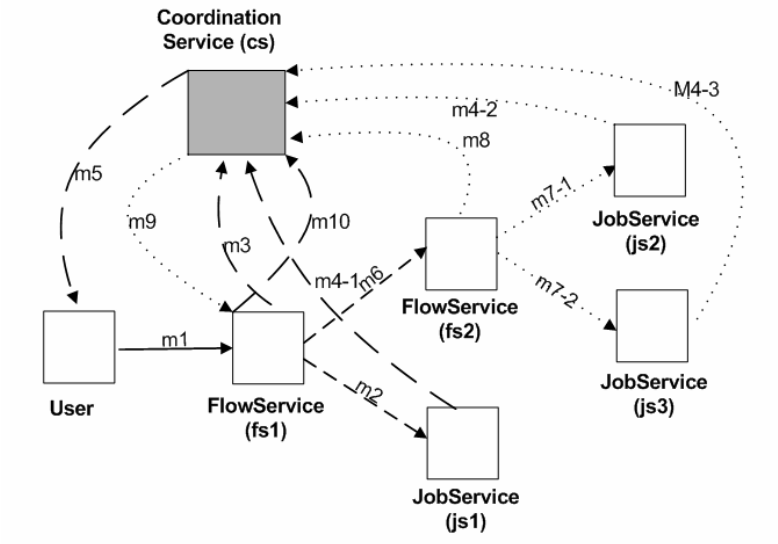


Fig. 5 A complex job submission example

Since all the interactions are coordinated by the Coordination Service and ContextStore, all interactions started before and after the re-configuration can be executed consistently and without interruption. Also, when replacing a service, an important requirement is that other services do not need to be aware of the change. Therefore, after the re-configuration, the FlowService fs2 needs to behave as a JobService when it receives jobs from fs1 (m6), and as a FlowService when it passes the sub-jobs to js2, js3 (m7) and forwards the job state to cs (m8).

Thus, two messagedefs need to be added to the FlowService specification: one describes how a FlowService (in our example, fs2) handles a jobexecute request, the other describes how a FlowService (fs1 in our case) handles a jobsreply message.

```
OnMessage [ to:FS, query:[ from:u,query:[ j1,j2],context:c], function:jobexecute ]
        generate new uid
        send [ to:CS, query:[ from:FS,query:[from:u,query:[ j1,j2],context:c],context:uid],
                      function:jobssubmitted ]
        send [ to:JS1, query: [ from:FS,query:j1,context:uid], function:jobexecute ]
        send [ to:JS2, query: [ from:FS,query:j2,context:uid], function:jobexecute ]

OnMessage [ to:FS, query:[from:cs,query:[from:u,query:job,context:c],result:result,context:uid],
                      function:jobsreply ]
        send [ to:CS, query:[ from:FS,query:job,result:result,context:c], function:jobcomplete ]
```

Fig. 6. Updated FlowService Specification

The first messagedef is similar to startjobs in Fig. 2, but with a different function name. This time, the user, u, of fs2 is a FlowService, fs1. When fs2 receives a

jobexecute message, it will start a new interaction by creating a new identity. As a result, the Coordination Service will create two interaction records in the ContextStore, for fs1-submitted jobs and fs2-submitted jobs (Table 1). From the point of view of the Coordination Service, fs1 is the user of fs2-submitted jobs; thus, the Coordination Service will send replies to fs1 corresponding to those jobs.

The second messagedef specifies that, when fs1 receives a jobsreply message from the Coordination Service, it extracts the query and informs the Coordination Service that the jobs previously submitted to fs2 have been completed. A jobcomplete message received by the Coordination Service from a FlowService produces the same result as one received from a JobService.

Table 1. Interactions State at ContextStore

ContextStore[uid1]	ContextStore[uid2]
[from: fs1, **query: [from:** u, **query: [** j1,**[** j2,j3**]],** **context:** c**]],** **[from:** js1, **query:** j1, **result:** r1**],** **[from:** fs2, **query: [** j2,j3**], result:** [r2,r3**]]**	**[from:** fs2, **query: [from:** fs1, **query: [** j2,j3**],** **context:** uid1**]],** **[from:** js2, **query:** j2, **result:** r2**],** **[from:** js3, **query:** j3, **result:** r3**]**

This example demonstrates the capability of using DFM to model long-running interactions and dynamic configurations. By sharing interfaces, a service behaves multifunctionally. Assuming that the FlowService fs1 is able to handle a jobsreply message, this service doesn't have to be stopped and rewritten in order to allow the JobService js2 to be replaced by a FlowService fs2 (that is able to handle a jobexecute message). Also, using our context and coordination mechanisms, a query can be completed by two hierarchical interactions (Table 1). Therefore a service can not only be replaced by a peer service, but also by a service with more workflows.

We also note that our query structure and use of contexts is highly flexible in terms of the kinds of service-oriented applications it can model. An application coordinated by more than one service could, for instance, be modeled with a less hierarchical query structure than in our previous example. Applications involving service compositions that do not require either a CoordinationService or a ContextStore can also be described. In addition to modelling service compositions, our context and coordination mechanisms can also be used to model business processes in specific domains, or security sensitive applications.

5 Conclusion

A service-oriented application is composed by dynamic services orchestrated using asynchronous messages. We have introduced a formal modelling notation which uses context and coordination mechanisms to specify asynchronous web services composition, and has the additional capability to support long-running interactions and dynamic configurations. An operational semantics for this notation has already been developed, see [6] for details. Future work includes the use of this operational semantics to develop a simulation tool for asynchronous web services coordination.

References

1. Booth, D. et al.: Web Services Architecture, http://www.w3.org/, 2004.
2. Peltz, C.: Web services orchestration and choreography, IEEE Computer, vol. 36, 2003, pp. 46-52.
3. Bunting, D. et al.: Web Services Composition Application Framework (WS-CAF) V1.0, http://www.oasis-open.org, 2003.
4. Andrews, T. et al.: Business Process Execution Language for Web Services Version 1.1, http://www.ibm.com/developerworks/library/ws-bpel/, 2003.
5. Henderson, P. and Yang, J.: Reusable Web Services, Proceedings of 8th International Conference, ICSR 2004, Madrid, Spain, 2004, pp. 185-194.
6. Yang, J., Cîrstea, C. and Henderson, P.: An Operational Semantics for DFM, a Formal Notation for Modelling Asynchronous Web Services Coordination, accepted by First International Workshop on Services Engineering (SEIW 2005), Melbourne, Australia, 2005.
7. Christensen, E. et al.: Web Services Description Language (WSDL) 1.1, http://www.w3.org/TR/wsdl, 2001.
8. Henderson, P.: Laws for Dynamic Systems, Proceedings of International Conference on Software Re-Use (ICSR 98), Victoria, Canada, 1998.
9. Henderson, P.: Modelling Architectures for Dynamic Systems. In: McIver, A. and Morgan, C. (eds.), Programming Methodology, Monographs in Computer Science, Springer, 2003.

Realising Personalised Web Service Composition Through Adaptive Replanning

Steffen Higel, David Lewis, and Vincent Wade

Knowledge and Data Engineering Group,
Department of Computer Science,
Trinity College Dublin, Ireland
{higels, delewis, vwade}@cs.tcd.ie

Abstract. The emergence of fully-automated Web service composition as a potential facilitator of both eBusiness and ambient or ubiquitous computing is to be welcomed. However this emergence has exposed the need for flexibility and adaptivity due to the fundamentally unreliable nature of the networks and infrastructure on which the component services rely. Furthermore, a key to driving forward acceptance and adoption of this growing set of technologies is the improvement of the user's overall experience therewith. Our experimentation has proven that it is quite possible to generate inflexible and only partially adaptive service compositions using out of the box A.I. planners. Because modifying the planner is beyond the scope of our research, we seek to use methods of pre-processing and post-analysis to enable AI planners to produce adaptive compositions. In this paper, the current state of our research is presented along with a proposed direction for improving the reconciliation of user needs with the available services.

1 Introduction

The composition of software components into larger pieces of useful software is a remarkably simple concept. We feed the outputs of one component into the inputs of one or more others in the anticipation that the net effect will be close to what we want to achieve. When coupled with semantics supported by ontologies, the mature manageability and audit ability found in work-flow and the power of A.I. planning, it should become possible to automatically generate robust and personalised compositions at near run-time in response to a user's needs.

Web services and their associated technologies have become an enabler of "loosely coupled, dynamically bound components" [Eisenberg, 2001]. We and others [Kocoman et al, 2001] believe that these components can be dynamically bound using completely automated processes. Furthermore, these processes can be tailored around a user using techniques developed in the areas of adaptive hypermedia [Conlan et al, 2003].

In this paper, we will present how we automatically create compositions using A.I. planners, how these compositions are then analysed for weaknesses and how we then recreate sections of the plans to provide the user with a more

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 49–58, 2005.
© Springer-Verlag Berlin Heidelberg 2005

personalised experience. The findings of our experimentation are presented and we then discuss the research that needs to be carried out before we can achieve our vision of how adaptive, personalised service composition can take place.

2 Motivations

A Web service, that is a software component that uses technologies originally found on the World-Wide-Web for communication and self-description [Scott, 2001], provides an open and lightweight means with which users and other pieces of software can interact. Chaining these services together by feeding the outputs of one into the inputs of one or more other Web services is a simplistic but fundamentally correct view of what Web service composition is. Because the dynamic linking of these software components is so easy to achieve (purely a run-time process), with a sufficiently large volume of services available, the generation of tailored software which at least corresponds to the functional requirements of a user or entity is entirely feasible. When we have a richer set of descriptive mechanisms both on the non-functional properties of the service and the requirements and preferences of the user, it becomes possible to customise the composition very specifically to the user [Higel et al, 2003]. This adaptivity opens an array of possibilities in many established and emerging areas of computing. We believe that the two most significant ones are those of ubiquitous computing and eBusiness.

In the domain of ubiquitous computing, where intermittent and location dependent availability of services becomes a factor, a more flexible approach to user interaction with the environment and services is desirable. Web services, with their openly defined (and arbitrarily verbose) interfaces provide a suitable encapsulation of environmental functionality [Issarny et al, 2005] and through the support of ontologies offer ample means of abstracting Web services so that they can be resolved to instances providing the correct functionality at run-time. As users begin to require more complex interactions with their surroundings, service composition can be used to create and manage these requirements.

As various high-profile organisations begin to provide some of their business functionality via Web services (specifically SOAP and WSDL)[1], it is our belief that through appropriate semantic support, the next generation of online commercial interactions may be entirely facilitated by Web service composition. Businesses which realise their core competencies can focus purely on providing the parts of a business interaction at which they excel and allow others to do the same with other parts of said interaction.

In both of these research areas, an improved user experience will lead to improved user acceptance. We believe that transparency and adaptivity are a key to driving this. To facilitate this these, we require a rich means of expressing the needs of a user and any other requirements that might be imposed by their environment or any entity that they are representing.

[1] See http://soap.amazon.com/schemas2/AmazonWebServices.wsdl and http://www.google.com/apis/ for two prominent examples.

3 Existing Research and Technologies

3.1 Automatic Service Composition

Automatic Service Composition, the generation of chains of services by software, has largely grown from the application of established knowledge engineering and work-flow management techniques to those of artificial intelligence. Service composition based on incremental planning, hierarchical task network planning and graph theory all achieve their tasks effectively with varying degrees of automation. Languages used to describe service compositions like BPEL4WS and WSCL have evolved from work-flow description languages.[van der Aalst et al, 2003]. As previously mentioned, the use of adaptive hypermedia techniques can do much to guide research into the tailoring service compositions to individual users.

What we are presently lacking is a general mechanism for describing how to aggregate and reason on the non-functional properties of services and how to analyse the suitability of a chain of services when compared to a user's needs and requirements. Work has been done to solve specific portions of this problem, for instance [Jaeger et al, 2004] goes into some depth on how to model and aggregate different sub-concepts in quality of service while [Quinn et al, 2005] explores the same idea in the realm of trust and security. It should however be clear that a general mechanism for expressing arbitrary aggregations would be preferable, rather than relying on specific and differing rule-sets for each class of non-functional property.

3.2 A.I. Planning

Artificial Intelligence Planning creates sequences of pre-defined actions to alter a given set of environmental entities from one supplied set of states to another desired set of states. The description of these actions consists of a set of required states (for instance a book-selling action might require that a given person actually wants to buy a book), a set of resultant states (using the previous example, the action might result in the required book being ready to ship to a user).

The Problem Domain Definition Language (currently at version 2.2) is used to describe two general concepts to a planner[2]. The first is the domain description, which outlines the available actions, their preconditions and outputs and other configuration options that the planner should use. The second is the problem specification, which contains a list of all of the instances of objects that are to be considered when creating the plan. More importantly, the problem specification supplies the planner with a list of current environmental states and a list of desired environmental states (see Fig. 1.). Upon being executed, the planner will search through the available actions for a path which leaves the environment in the desired state, iteratively improving the plans until either the search space is exhausted or it exceeds the alotted time.

[2] See http://users.raise.anu.edu.au/~thiebaux/workshops/ICAPS03/ for a number of publications from the ICAPS Workshop on PDDL, 2003.

```
(:init (haspaymentconfirmation companyd moneyamounte personb ) (readyto-
ship book
titlec personb ))
(:goal(hasbook booktitlec personb ))
```

Fig. 1. PDDL representation of a user's initial state and goals

An evaluluation of the planners which competed in the 2004 International Planning Competition eventually brought us to use LPG-td [Gerivini et al, 2004], which has full support for the two new requirements for PDDL2.2 compliance (derived predicates and timed initial literals, neither of which had any particular bearing on our research). In terms of flexibility, it allowed us to very quickly experiment with different search algorithms, maximum search times and search depth.

4 Current Work

We have developed an API for generating PDDL, analysing the output of the planner and affecting the "rankings" of the available services to reflect user preferences. In this section, the architecture and processes involved in using these pieces of software is discussed.

4.1 Architectural Overview

From a component-oriented perspective, our architecture at present consists of a service repository, a PDDL generator, a planner, a planner output analyser and a replanner-controller. The service repository is a relational database which stores the functional and non-functional properties of the available services. It is our intention to replace this with an XML database which will store OWL-S documents, as certain properties of relational databases make this type of work somewhat cumbersome. There is a lack of openly available, composable service descriptions, so we have developed a set of our own service descriptions using the by now customary "buying a book" example. Of note is our treatment of the concept of a book. We treat an electronic representation of a book as being equivilant to that of a "dead-tree" copy of the book.

4.2 PDDL Generation and Basic Adaptivity

It should be quite apparent that there is no one-to-one mapping between each concept in Service Composition and A.I. planning. Planning deals purely with the functional and practical requirements for and outcomes of executing an action. Research into service composition has developed a richer set of descriptive mechanisms for expressing properties beyond those which are functional. What mapping can be done is relatively straightforward (though the same cannot be said for the actual generation of syntactically correct PDDL!).

We use PDDL's durative actions to represent each service from our repository in the PDDL domain representation for reasons that shall be outlined later

in this section. The list of types and predicates, required as part of the domain representation are all derived and sanity-checked from the available actions provided by each service. The PDDL problem representation generator is fed a list of predicates and goals required by the user which are converted into an appropriate form (see Fig. 2.).

```
(:predicates
(wantsbookgenre ?a - bookgenre ?b - person)
(wantsbook ?a - booktitle ?b - person)
(owesmoney ?a - company ?b - moneyamount ?c - person)
(haspaymentconfirmation ?a - company ?b - moneyamount ?c - person)
(readytoship ?a - booktitle ?b - person)
)
```

Fig. 2. PDDL representation of predicates, required as part of the domain representation

PDDL durative-actions differ from other PDDL action representations by allowing the PDDL author to specify the duration of the action's execution time (see Fig. 3.). Because a planner, when presented with two actions of differing durations but identical functional properties, will prefer that with the lower duration, we are presented with an opportunity to perform limited adaptivity. In essence, part of our experimentation involves the distillation into a single number of the user's preferences and any other relevant policies applied to the available services, allowing the planner to create an adapted and optimal plan.

```
(:durative-action fastbookrecommendation
:parameters (?a - bookgenre ?b - person ?c - booktitle )
:duration (= ?duration 71)
:condition (at start(wantsbookgenre ?a ?b ))
:effect (at end(wantsbook ?c ?b ))
)
```

Fig. 3. PDDL representation of a book recommendation service

4.3 Service Classification

A reduction of the search space in any planning problem will decrease the execution time of the planner. The abstraction of service functionality back to classifications based on their functional properties allows us to delay committing to using a given service until near-execution time of the composition. These two observations are what motivates our use of a service classifier. This analyses a set of services and groups them by like functional properties. As an aside, it should be noted that in a real-world environment, this would require some ontology mapping support to bridge gaps in the terminology used by two different service providers. Instead of feeding "raw" service descriptions into the planner, we instead feed classifications which can be then resolved at a later date, based on both service availability and user requirements.

4.4 Instantiating the Planner, Interpretting Its Output, Performing Adaptivity

The planner is fed a problem file, a domain definition and various parameters governing the nature of its execution (search type, maximum execution time, search depth, verbosity etc.). We simply fork the process and wait for it to complete its execution. The planner prints out a list of all of the solutions it has produced, along with the files in which it has stored these (see Fig. 4.). The output is parsed and the string representations of the service classes or services are resolved back into the internal representations of those service classes within the software.

The service classifications must then be resolved back into appropriate services, based on supplied user requirements. At present, we have a simplistic, normalised representation of user non-functional properties which are placed into a matrix and multiplied by a similar matrix derived from the available services. An average is generated from the elements of the resultant matrix which is then used as a suitability metric for this service. The highest scoring service is selected and placed into the appropriate point in the solution. This simplistic representation of user preferences and service non-functional properties is a stop-gap, whose replacement is outlined in the *Further Work* section.

```
0.0003: (CHEAPBOOKRECOMMENDATION BOOKGENREA PERSONB BOOKTITLEC) [76.0000]
76.0005: (FASTBOOKSELLING BOOKTITLEC PERSONB COMPANYD MONEYAMOUNTE)
[78.0000]
154.0007: (FANTASTICPAY COMPANYD MONEYAMOUNTE PERSONB BOOKTITLEC)
[82.0000]
236.0010: (WONDERDELIVERY COMPANYD MONEYAMOUNTE PERSONB BOOKTITLEC)
[75.0000]
```

Fig. 4. An example of a solution file generated by the planner

4.5 Replanning

If the resolution of a service classification back to an existing service provides a poor match, it might be possible to generate a plan which is longer (i.e. consists of more services) but which in a non-functional sense, is more closely matched with what the user desires. A user might prefer to have a PDF emailed to them rather than a "dead-tree" copy sent through the post, even if the composition involves invoking a larger number of services. Assuming our descriptions of non-functional properties can expose the shortcomings of the shorter solution, our software is capable of removing the offending classification from its list of services and can then attempt to generate a new plan to bridge this gap in our overall service composition.

In our experimentation, we created two possible paths through a given composition. A book could effectively be purchased in "dead-tree" format and shipped as a parcel through the postal service or as an eBook that would be converted

into a PDF and then emailed to the user. By modifying the user's preferences, our planner-output interpretter would identify a potential weakpoint in the plan and then instruct the planner to attempt to generate a new solution for that portion of the composition.

4.6 Experimentation

Given that we want to find out how best to adapt based on the information expressed in the services' non-functional properties and the user's view thereof, we have attempted to base our experimentation around the following questions:

- Is the classification of services before the planning process and then replanning the weakpoints the most logical technique for maintaining optimal adaptivity (see Fig. 5.)? Or can we leave all adaptivity down to the planner using the above techniques (see Fig. 6.)? This was examined by running the composition process repeatedly and measuring its execution time and comparing the suitability of the composition to the user each time. This appropriateness is examined by comparing each constituent service to its functionally equivilant peers and examining if it is the one best suited to the user. Because we lack a proper means of aggregating and comparing chains of services of different lengths, for now, this is the only test we can perform.
- Can the modification of a user's preferences completely alter the resultant plan, for instance resulting in a plan which emails a PDF version of the

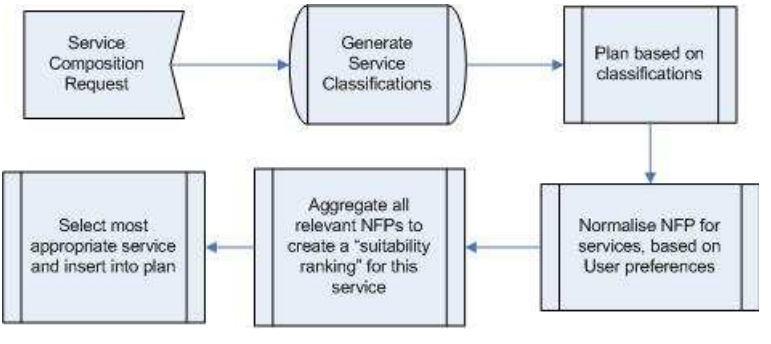


Fig. 5. Service Composition using pre-classification and post-analysis

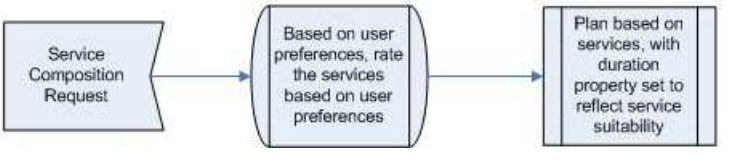


Fig. 6. Service Composition using pre-rating of services

requested book to a user rather than shipping a copy of the book through the post? This could be termed taking a different *functional path* through the solution space.

The experiment was carried out on an 800Mhz Pentium 3 running Linux. The wrapper and APIs around the planner were written in Python. Our pool of initial services consisted of 62 distinct services, which could be classified into 7 different classes. The experiment was run 1000 times for each approach.

5 Results

5.1 Classification vs. Pre-rating

Our experimentation with the pre-rating technique showed that over 1000 executions, the average time to generate the composition was 1.969 seconds and had a 100% match rate to a composition pre-determined to have been the closest match to a user's preferences, as best as this can be defined at this point. Classification and post-planning resolution, over the same number of executions, averaged 2.103 seconds and again, had a 100% accuracy rate. The difference between the two is barely discernable, so it stands to reason that we can evaluate both techniques on what we believe to be important: their conduciveness to user-centric-adaptivity.

5.2 Plan Modification

We modified the supplied user preferences to quite blatantly express their desire for low-cost solutions at the expense of reliability, security and latency. The functional path which resulted in a PDF version of the book being emailed was set to be practically free. In this case, the planner did select plans based on this path as being the most suitable option. When using the classification and replanning approach, the replanner spotted a potential weak-point in the plan and replaced it with the same PDF generating composition.

However, it is simple to represent cost as a durative property that the planner understands. Non-functional properties which cannot simply be aggregated by adding them together cannot be treated in the same way, so relying on the planner to perform this adaptivity will never provide a general solution. Even if we intelligently distill all non-functional properties into a single suitability metric, without severely skewing the *duration* properties of the actions, which will require hand-tuning on a case by case basis, the planner will never have sufficient understanding of the user's preferences such that it might generated a longer but more suitable plan. For this reason, it is only through analysis of weakpoints within the plans and replanning based on the functional properties of that point that we can adapt and refine the compositions on a more "informed" level.

It is this exact idea that we intend to explore further.

6 Research Directions and Further Work

It became quite clear throughout the implementation of this experiment that simple normalised representations of a user's view of the importance of a given non-functional property or the service's description of its own non-functional properties are insufficient. In reality, we require an appropriate means of describing how to manipulate and reason on non-functional properties for the adaptivity and flexibility to be useable in a real-world scenario. We have identified three primary types of non-functional property aggregation:

- When two different users refers to a non-functional property like "cost", they might imply different things. For one, the initial outlay might be extremely relevant, for the other, the total cost of ownership would play a larger part in this definition. If the Web service defines both of these, we should allow the user to specify a ruleset for aggregating seemingly irrelevant NFPs for the service into ones that are meaningful to them. Once this is done, the planner and analysis software can properly make decisions on how to compare two given services.
- A service could be described in terms of its cost, reliability or security. When aggregating these together to form a single suitability metric for a service, we may find that some non-functional properties conflict or need to be aggregated in unique ways. A non-functional property might be represented non-numerically (a service might only be available on "Wednesday") and we intend on providing a means of expressing the relevance (or lack there of) of such properties on the overall suitability of a service.
- When attempting to analyse the effectiveness of a chain of services based on user preferences, we are presented with the simple reality that aggregating like NFPs across multiple services requires rulesets defining how this should occur. To aggregate the cost of using a set of services is trivial: we add the cost of each service and compare it to another chain to find the cheapest. However, when analysing the security of such a chain, we might take the value of the weakest link. The mean time between failures might be calculated as an average of all of the services being used. For this reason, a descriptive mechanism is required to aid these decisions.

7 Conclusions

In this paper, we have outlined the techniques we have used to generate trivial service compositions with limited adaptivity using an incremental planner. We have demonstrated why we feel pre-composition service classification is conducive to increasing the adaptivity of the composition itself. Also, we have deduced that replanning is key to creating refined and personalised compositions, and that having a suitable means for describing appropriate aggregation techniques for the non-functional properties of services is critical to adapting service compositions to a user's specifications.

References

[Eisenberg, 2001] "Preparing for the Web Services Paradigm", Eisenberg, B., W3C workshop on Web Services, 2001

[Kocoman et al, 2001] "Towards Zero-Code eService Composition", Kõcõman E., Melloul L., Fox A., Hot Topics in Operating Systems, 2001

[Conlan et al, 2003] "Applying Adaptive Hypermedia Techniques to Semantic Web service Composition", Conlan O., Lewis D., Higel S., O'Sullivan D., Wade V., Adaptive Hypermedia 2003

[Scott, 2001] "The Road to Web Services", Scott, G., W3C workshop on Web Services, 2001

[Higel et al, 2003] "Towards an Intuitive Interface for Tailored Service Compositions", Higel, S., O'Donnell, T., Lewis, D., Wade, V., 4th IFIP International Conference on Distributed Applications & Interoperable Systems, 2003

[van der Aalst et al, 2003] "Web Service Composition Languages: Old Wine in New Bottles?", van der Aalst, W.M.P., Dumas, M., ter Hofstede, A.H.M., 29th Euromicro Conference, 2003

[Jaeger et al, 2004] "QoS Aggregation for Web Service Composition using Workflow Patterns", Jaeger M. C., Rojec-Goldmann, G., Muhl. G., edoc, vol. 00, no. , pp. 149-159, Enterprise 2004.

[Quinn et al, 2005] "deepTrust Management Application for Discovery, Selection, and Composition of Trustworthy Services", Quinn, K., O'Sullivan, D., Lewis, D., Wade, V.P., 9th IFIP/IEEE International Symposium on Integrated Network Management (IM 2005), 2005

[Issarny et al, 2005] "Developing Ambient Intelligence Systems: A Solution based on Web Services", Issarny, V., Sacchetti, D., Tartanoglu, F., Sailhan F., Chibout, R., Levy, N., Talamona, A., Automated Software Engineering, Volume 12, Issue 1, 2005

[Gerivini et al, 2004] "LPG-TD: a Fully Automated Planner for PDDL2.2 Domains", Gerevini, A., Saetti, A., Serina, I., International Planning Competition, 14th Int. Conference on Automated Planning and Scheduling (ICAPS-04), 2004.

Semantic Web Services Discovery in Multi-ontology Environment

Sasiporn Usanavasin[1], Shingo Takada[1], and Norihisa Doi[2]

[1] Graduate School of Science and Technology, Keio University, Yokohama, Japan
zim@doi.ics.keio.ac.jp, michigan@ics.keio.ac.jp
[2] Faculty of Science and Engineering, Chuo University, Japan
doi@keio.ac.jp

Abstract. Web services are becoming the basis for electronic commerce of all forms. The number of services being provided is increasing but different service providers use different ontologies for services' descriptions. This has made it difficult for service discovery agents to compare and locate the desired services. Inputs and outputs are important pieces of information that can be used when searching for the needed services. Therefore, in this paper, to facilitate users or software agents for discovering Web services in multi-ontology environments, we propose an approach to determine the semantic similarity of services' inputs/outputs that are described by different ontologies.

1 Introduction

Web services are becoming the basis for electronic commerce of all forms. Companies invoke the services of other companies to accomplish a business transaction. In an environment in which only a few companies participate, managing the discovery of business partners manually would be simple. However, the increase in the number of participating companies and the number of available services have made it difficult for users or software agents (service discovery systems) to compare and retrieve the needed services. UDDI [1] exists for this reason. UDDI provides a platform-independent way of describing and discovering Web services. UDDI data structures provide a framework for describing basic service information, and detailed service access information using WSDL. However, WSDL only provides information on the services' functionalities at the syntactic level without any formal definition to what the syntactic definitions might mean. Unless the client agent knows the exact form and meaning of a service's WSDL in advance, the combination of UDDI with WSDL and coarse-grained business descriptions is not enough to allow fully automated service discovery and usage.

In the past few years, the Semantic Web [2] has been introduced to overcome the syntactic problems by adding semantic meaning to Web-based information. The Semantic Web is an extension to the Web technologies in which information is given explicit meaning, making it easier for machines to semantically and automatically interpret and process information available on the Web. These

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 59–68, 2005.
© Springer-Verlag Berlin Heidelberg 2005

explicit meanings are described through what is called *ontology* [3]. In Web services, ontologies are used to describe service functionalities and restrictions including input and output interfaces.

Current Web services are created by many different parties and each party tends to use its own ontology to describe its services, making the search process difficult. Previous attempts to develop automatic Web service retrieval systems incorporating Semantic Web technology include [1,4,5,6]. One of the main focuses of these studies is to discover Web services by matching descriptions of input and output parameters between the request service and the existing service advertisement profiles. But their studies are based on a single shared ontology, which is an impractical assumption. Thus, a more efficient and intelligent system is needed and this system should be able to handle the issue of semantic similarity between different ontologies that are used to describe those available Web services.

In this paper, we present a way to facilitate the service discovery system by presenting an approach to determine the semantic similarity of services' inputs/outputs in a multi-ontology environment.

This paper is structured as follows. Section 2 discusses related works. Section 3 presents our approach to calculating semantic similarity between inputs/outputs of Web services from different ontologies. Section 4 shows our evaluation, and Section 5 presents our conclusions and future work.

2 Related Works

One way to support service discovery agents in a multi-ontology environment is to calculate the semantic similarity of information being described by different ontologies. This mechanism is based on an idea similar to what is used for ontology merging. In ontology merging, entities of different ontologies are compared based on syntactic and semantic relations in order to locate similar entities that will be used as the merging points to form a larger, more general ontology. One of the first tools was developed by Hovy et al. [7]. Their tool is mainly based on using lexical information, such as concept names, definitions, lexical structure, and distance between strings.

Rodríguez et al [8] presented an approach for computing the semantic similarity of entity classes by using a matching process over synonym sets, semantic neighborhoods, and distinguishing features that are classified into parts, functions, and attributes. They compared distinguishing features in terms of a strict string matching between synonym sets that refer to those features. Their approach is focused on the spatial domain and greatly depends on the representation of the entity classes.

Noy and Musen developed systems for performing ontology merging and alignment in the Protégé-2000 ontology development environment, which are known as PROMPT [9] and its successor Anchor-PROMPT [10]. These tools determine whether entities from the two ontologies have similar labels and then suggest the merging points between the two ontologies based on entities that have similar labels. Anchor-PROMPT represents an advanced version of PROMPT by

including similarity measures based on ontology structure. However, the ontology structure is considered only for determining the similarity of concept classes. For other types of entities, only label similarity is considered.

Ehrig et al [11] proposed an approach that uses a number of similarity measurements to identify the mapping of entities between two ontologies. The similarity between two entities is an aggregate of similarity measures such as labels, sub/super concepts, sub/super properties, instances, etc. Though this approach shows high quality results, the run-time complexity increases exponentially with large-sized ontologies. The long response times make it impractical to apply this approach for a service discovery system, where more than two ontologies are needed to be considered at a time.

None of the above works are suitable and practical for our current objective. Since our goal of computing semantic similarity between different ontologies is to resolve the heterogeneity of Web services during the discovery process, both quality and response time have to be considered.

3 Our Approach for Calculating Cross-ontology Similarity of Web Services' Inputs/Outputs

Web service discovery is the process of locating a service that has a desired functionality. Inputs and outputs are important pieces of information that can be used when searching for the needed services. Since different service providers will likely use different ontologies to describe their services, we need a mechanism that can help evaluate the similarity of those services' information. We propose an approach for calculating the semantic similarity of Web services' inputs/outputs that are described by different ontologies.

Our approach is based on the following definitions. An ontology $\Omega_i = \{c_1, \ldots, c_n\}$ contains a set of concept classes. Each concept class (or domain class) has a name or label and has an associated set of properties $P_j = \{p_1, \ldots, p_m\}$. Each property has a name and a range class that indicates the data type the property can take. For example, a concept class named *Book* may have properties named *bookTitle* and *ISBN* and these properties may take string as their data types. Thus, inputs or outputs of a Web service for an online book store can be described by these properties. When comparing inputs/outputs from two services that are described by two different ontologies, we refer to the first service as the source service (W_S) and the second service as the target service (W_T). Since both inputs and outputs of services are described by properties within ontologies, we refer to them as properties (the source property (p_S) and the target property (p_T)). In this work, we focus on calculating the similarity between properties from different ontologies. Specifically, the similarity between two properties is calculated based on the following information:

- URIs
- names or labels
- a set of synonyms

- range classes (classes that define data types)
- domain classes (classes that the properties are associated with)
- superclasses (parent classes of the domain class)
- subclasses (child classes of the domain class)

3.1 URI Similarity

Uniform Resource Identifiers (URIs) are strings that identify resources on the Web and they are unique for each object. We know that if two inputs/outputs have the same URI, they must be identical. Therefore, we start by checking the similarity between the source property $uri(p_S)$ and the target property $uri(p_T)$. If they have the same URI, i.e. $uri(p_S) = uri(p_T)$, we conclude that the two properties are identical. Otherwise, other similarity evaluations have to be performed to determine if they are semantically similar to each other.

3.2 Name Similarity

The next simplest way to determine the similarity of two properties is by comparing their names or labels ($n(p_S)$ and $n(p_T)$). The name similarity function, $NSim(n(p_S),n(p_T))$, is a function that calculates similarity between two strings and is defined as follows.

$$NSim(n(p_S),\, n(p_T)) \;=\; \frac{\max\,(n(p_S),\, n(p_T))\text{ - LD}}{\max\,(n(p_S),\, n(p_T))} \tag{1}$$

This function is computed based on the well-known technique for approximate string matching, which is called Levenshtein Distance (LD) [12]. The Levenshtein distance is the number of deletions, insertions, or substitutions required to transform $n(p_S)$ into $n(p_T)$, as defined below.

$$\text{LD} = \text{Deletion} + \text{Insertion} + \text{Substitution} \tag{2}$$

3.3 Synonym Similarity

Even though string matching can help specify how close two strings are, it is not always the right indicator because there are cases where the distance between two strings is small but they have very different meaning. Therefore, we also consider synonyms through which we fine-tune the overall similarity values such that the system can indicate the more correct results. We use the WordNet [13] ontology as a reference for synonyms. We define a synonym similarity function (*SynSim*) as a function for calculating similarity between synonym sets of two entities. The maximum value of synonym similarity is 1 when two entities have the same names (NSim = 1). Otherwise, the *SynSim* function (i.e. between two properties) is calculated as follows.

$$SynSim(p_S, p_T) = \frac{|(Syn(p_S) \cap Syn(p_T)|}{|(Syn(p_S) - Syn(p_T)) \cup (Syn(p_T) - Syn(p_S))|} \tag{3}$$

where $Syn(p_i)$ is the set of synonyms for p_i.

3.4 Data Type Similarity

Each property has a range that indicates the type of data it can take. In our study, we have chosen the built-in data type hierarchy defined in XML Schema [14] because it provides a standard and comprehensive data type hierarchy and most of the existing ontologies refer to this hierarchy. The similarity of the data types or ranges of p_S and p_T is evaluated using the function $TSim(r(p_S), r(p_T))$ which takes a value as defined below.

$$\begin{array}{ll}
1 & if\ r(p_S) = r(p_T) \\
1 & if\ r(p_S) = anytype, \quad r(p_T) = string \\
1 & if\ r(p_S) = integer, \quad r(p_T) = float \\
2/3 & if\ r(p_S) = float, \quad r(p_T) = integer \\
1/3 & if\ r(p_S) = double, \quad r(p_T) = integer \\
1/2 & if\ r(p_S) = date, \quad r(p_T) = dateTime \\
1/2 & if\ r(p_S) = time, \quad r(p_T) = dateTime \\
1 & if\ r(p_S) = dateTime, r(p_T) = date \\
1 & if\ r(p_S) = dateTime, r(p_T) = time \\
0 & otherwise
\end{array} \tag{4}$$

The function $TSim(r(p_S), r(p_T))$ is formulated based on the competence of making data type conversions. The similarity takes the maximum value of 1 when the two properties have the same data type or the source property can be converted to the target property. The similarity value is less than 1 when the data type conversion is not preferred or recommended since a loss of information may occur. Note that we do not claim to cover all possible data type conversions. New types or other possible values can be added to the function *TSim* as needed.

3.5 Domain Class Similarity

The semantic similarity of the domain classes (denoted as c_S and c_T) of the two properties is calculated using function $DSim(c_S, c_T)$, which is based on two similarity values: the similarity of class names ($n(c_S)$ and $n(c_T)$) and the similarity of all properties that are associated to the domain classes (sometimes called feature similarity or attribute similarity). We give equal weights to the class name similarity and the feature similarity because we consider them as equally important.

$$DSim(c_S, c_T) = \frac{CNSim(c_S, c_T) + PSim(c_S, c_T)}{2} \tag{5}$$

The class name similarity, $CNSim(c_S, c_T)$, is calculated as shown in Eq. 6 based on the string matching and synonyms similarity.

$$CNSim(c_S, c_T) = \frac{NSim(n(c_S), n(c_T)) + SynSim(c_S, c_T)}{2} \tag{6}$$

The feature similarity, $PSim(c_S, c_T)$, is determined based on Tversky's model [15] as shown in Eq. 7.

$$PSim\ (c_S,\ c_T)\ =\ \frac{|PS \cap PT|}{|PS \cup PT|} \tag{7}$$

where PS and PT denote the set of properties that are associated with c_S and c_T respectively.

Tversky introduced a feature-contrast model that determines the similarity of two objects based on the number of common and unique features. When calculating the intersection of two sets of properties, two properties intersect (match) if the summation of their syntactic similarities, using Eq. 1 and 3, is greater than a threshold k. The value of $DSim(c_S, c_T)$ ranges from 0 to 1.

3.6 Neighborhood Concepts Similarity

In our approach, neighborhood concepts are superclasses ($\sup(c_S)$, $\sup(c_T)$) and subclasses ($\mathrm{sub}(c_S)$, $\mathrm{sub}(c_T)$) of c_S and c_T. We determine the similarity of neighborhood concepts according to Eq. 8. We calculate the similarities of the super and subclasses (Eq. 9 and Eq. 10) based on class name similarity and feature similarity (similar to Eq. 6 and Eq. 7). The value of $NBSim(c_S, c_T)$ ranges from 0 to 1.

$$NBSim(c_S, c_T) = \frac{\frac{\sum ParentSim(c_S, c_T)}{|matchedParent|} + \frac{\sum ChildSim(c_S, c_T)}{|matchedChild|}}{2} \tag{8}$$

$$ParentSim(c_S,\ c_T)\ = \frac{CNSim(sup(c_S), sup(c_T)) + PSim(sup(c_S), sup(c_T))}{2} \tag{9}$$

$$ChildSim(c_S,\ c_T)\ = \frac{CNSim(sub(c_S), sub(c_T)) + PSim(sub(c_S), sub(c_T))}{2} \tag{10}$$

3.7 Total Similarity Between Two Properties

The overall similarity between two properties is determined based on all similarity functions that are described from sections 3.1-3.6. For each pair of p_S and p_T, we first start by determining the similarity of their URIs. If they are identical, it means p_S matches p_T and the system will stop comparing for this pair and start the comparison for the next pair. Otherwise the system continues with the other similarity functions. The total similarity function is shown in Eq.11.

$$TotalSim = \omega_1 NSim + \omega_2 SynSim + \omega_3 TSim + \omega_4 DSim + \omega_5 NBSim \tag{11}$$

ω_1, ω_2, ω_3, ω_4, and ω_5 are weights for each similarity function.

In this study, given a p_S we try to find a p_T that matches the p_S the most. We consider p_S *matches* p_T if their $TotalSim(p_S,\ p_T)$ has a value greater than a threshold h. For any p_S and p_T that have total similarity lower than the threshold, they will be regarded as *not match*. In cases where there are multiple p_T's that match the p_S, the p_T that has the highest total similarity value will be selected.

3.8 Degree of Compatibility

We define the *Degree of Compatibility* (DOC) as a measurement to determine how compatible one service is to another in terms of inputs and outputs similarity. The value of DOC is used for ranking Web services in the discovery process. We determine the DOC between WS_S and WS_T based on the total similarity function (Eq. 11).

The goal of the service discovery is to match a request service with the available services in the repository. In this process, we refer to the request (service) as the source service (WS_S) and each of the available services as a target service (WS_T). Therefore, we compare each input i_S of I_S (inputs of WS_S) with each input i_T of I_T (inputs of WS_T), and compare each output o_S of O_S (outputs of WS_S) with each output o_T of O_T (outputs of WS_T). Thus, DOC between WS_S and WS_T is calculated as shown in Eq. 12. In this equation, we use the number of inputs and outputs of the request (service) as the denominators because the request is what the user wants.

$$DOC = \frac{\sum_{a=1,b=1}^{a=|I_S|,b=|I_T|} \frac{TotalSim(i_{S_a}, i_{T_b})}{|I_S|} + \sum_{m=1,n=1}^{m=|O_S|,n=|O_T|} \frac{TotalSim(o_{S_m}, o_{T_n})}{|O_S|}}{2}. \quad (12)$$

4 Evaluation

We compare our work with Ehrig's integrated approach for ontology mapping [11]. The reason that we chose Ehrig's approach is that they use several similarity measures to determine the similarity between entities providing high quality results. Other tools such as PROMPT [9] and Anchor-PROMPT [10] mainly determine only label similarity, and Rodríguez's work [8] focuses on comparing entity classes that have the representation in terms of distinguishing features (parts, functions, and attributes), which is not compatible with our approach.

We focused on testing our approach by using a set of existing ontologies. We used six ontologies from two different domains: university & research and person & contact (Table 1). Table 2 shows characteristics of these ontologies.

We performed the similarity calculations for each data set with our model and with Ehrig's model. Data set A and B each has three ontologies. To calculate

Table 1. Ontologies for experiments

Abbreviation	URI
swrc1a	http://www.aifb.uni-karlsruhe.de/WBS/meh/mapping
agenda-ont	http://www.daml.org/2001/10/agenda/agenda-ont
academia	http://www.doi.ics.keio.ac.jp/~zim/work/ont/academia
Person1	http://orlando.drc.com/daml/ontology/Person/3.1/Person-ont
Person2	http://pervasive.semanticweb.org/ont/2004/01/person
Person3	http://daml.umbc.edu/ontologies/ittalks/person

Table 2. Characteristics of Ontologies

	Concept	Property	Instance	Sub/Superconcept	Sub/Superproperty
swrc1a (Set A)	√	√	√	√	√
agenda-ont (Set A)	√	√		√	
academia (Set A)	√	√		√	
Person1 (Set B)	√	√		√	√
Person2 (Set B)	√	√	√	√	√
Person3 (Set B)	√	√		√	

the similarity for each data set, we took one ontology as the source and the other as target. Then for each concept and property in the source ontology we used our calculation to obtain the best match in the target ontology. Then we checked the returned results from both models with the answers, which were created by manual matching.

Table 3 shows average response times, precisions, recalls, and F-measures for data sets A and B. The definitions of precision and recall are given in Eq. 13 and Eq. 14 respectively. Higher precision and recall values show better result.

$$Precision = \frac{|X \cap Y|}{|Y|} \tag{13}$$

$$Recall = \frac{|X \cap Y|}{|X|} \tag{14}$$

X is the set of correct similar input/output properties (i.e., the "answers") and Y is the set of similar input/output properties calculated by our model. The F-measure [16] is a harmonic mean of precision and recall weighted by α to show the degree of effectiveness.

$$F - Measure = \frac{\left(\alpha^2 + 1\right) * Precision * Recall}{\alpha^2 * Precision + Recall} \tag{15}$$

α is the relative importance given to recall over precision. In our work, we consider recall and precision as equally important, i.e. $\alpha = 1.0$.

Although our precision are lower than those of Ehrig's approach (average 4% lower), we had higher recall (average 9.8% higher), which result in higher F-measures (4.1%). The differences in the recalls mainly result from considering the synonym and data type similarities especially when comparing the ontologies that have little information about sub/super concepts, sub/super property

Table 3. Experimental Results for Data Sets A and B

DataSet	Approach	Total Response Time(Sec.)	Precision(%)	Recall(%)	F-Measure(%)
A	Our	28.5	70.6	69	69.8
	Ehrig's	46	74.5	54.6	63
B	Our	18	76.5	72.2	74.3
	Ehrig's	30	80.1	67	72.9

relations, and instances. If we consider the total response time, we can see that our approach took only about ½ of their response time for both data sets.

Moreover, we also consider the average time per property. Our approach took about 0.69 second, and Ehrig's approach took approximately 1.09 seconds. The time difference of 0.4 second might seem small when we consider for just one property but if we think about incorporating the cross-ontology similarity to the service discovery system which may involve the calculations of hundreds or possibly thousands of properties (inputs/outputs), this small difference can greatly affect the total response time of the system.

Although our approach does not show a huge difference in F-measures compared to Ehrig's approach, we can claim that our approach can provide satisfactory and comparable results especially when we consider the response times (both for the average total response time and the average time per property). Since our main objective of performing cross-ontology similarity is to use it as part of our service retrieval system, not only precision and recall but the response time is a very important factor.

5 Concluding Remarks

In this paper, we presented an approach to determine the semantic similarity of inputs/outputs of Web services between different ontologies that will be used to enhance the service discovery process. Our cross-ontology calculation is based on the use of string similarity, synonym similarity, data type similarity, domain class similarity, super and subclasses similarities as measurements to determine the overall similarity of inputs/outputs between Web services. The total similarity values are then used to establish the degrees of compatibility between those services. In this paper, we showed a comparison of our approach with another approach. The results show that our approach can provide comparable results in terms of F-measures and faster response times. Although in this paper, we focused on determining the semantic similarity of Web services' inputs/outputs, our approach can also be applied to other service information such as comparing services' functionalities and services' products.

In our previous work, we developed a Web service discovery system [6] that takes a multi-faceted approach for searching Web services. However, the system was based on using a single ontology which is unrealistic. We plan to enhance our Web service discovery system by adding the cross-ontology similarity calculation that we presented in this paper. Moreover, we plan to run more experiments and perform evaluation on the overall performance of our system.

References

1. The UDDI Technical White Paper, Internet:http://www.uddi.org/, 2000.
2. T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web", Scientific American, Vol. 284, No. 5, pp. 34–43, 2001.

3. T. Gruber, "A Translation approach to portable ontology specifications", Knowledge Acquisition, Vol. 5, No. 2, pp.199–220, 1993.
4. M. Klein and A. Bernstein, "Searching for services on the semantic web using process ontologies", The Emerging Semantic Web-Selected papers from the first Semantic Web Working Symposium , I. Cruz, S. Decker, J. Euzenat, and D. McGuinness, Eds, Amsterdam: IOS press, pp. 159–172, 2002.
5. M. Paolucci, T. Kawamura, T. Payne and K. Sycara, "Semantic Matching of Web Services Capabilities", International Semantic Web Conference (ISWC), Italia, 2002.
6. S. Usanavasin, T. Nakamori, S. Takada and N. Doi, "A Multi-Faceted Approach for Searching Web Applications", Online version:IPSJ Digital Courier, Vol.1 (2005), pp. 166–180, Paper version:IPSJ Journal, Vol.46, No.3, pp.816-830, Mar 2005.
7. E. Hovy, "Combining and standardizing large scale, practical ontologies for machine translation and other uses", The First International Conference on Language Resources and Evaluation (LREC), pp. 535–542, Spain, 1998.
8. M. A Rodríguez and M. J. Egenhofer, "Determining Semantic Similarity Among Entity Classes from Different Ontologies", IEEE Transactions on Knowledge and Data Engineering, Vol. 15, No. 2, pp. 442-456, March/April 2003.
9. N. Noy and M. Musen, "PROMPT: Algorithm and tool for automated ontology merging and alignment", The 17th National Conference on Artificial Intelligence (AAAI'00), Texas, USA, July 2000.
10. N. Noy and M. Musen. "Anchor-PROMPT: Using non-local context for semantic matching", The workshop on Ontologies and Information Sharing at IJCAI2001, Seattle, WA, 2001.
11. M. Ehrig and Y. Sure, "Ontology Mapping - An Integrated Approach", The First European Semantic Web Symposium (ESWS2004), Vol. 3053, pp. 76–91, Greece, May 2004.
12. I. V. Levenshtein, Binary Codes capable of correcting deletions, insertions, and reversals, Cybernetics and Control Theory, Vol. 10, No. 8, pp. 707–710, 1966.
13. G. Miller, R. Beckwith, C. Fellbaum, D. Gross and K. Miller, "Introduction to WordNet: An On-Line Lexical Databases", International Journal of Lexicography, Vol. 3, No. 4,pp. 235-244, 1990.
14. P. V. Biron, K. Permanente and A. Malhotra, XML Schema Part 2: Datatypes Second Edition, October, 2004. Internet: http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/
15. A. Tversky, "Features of Similarity", Psychological Review, Vol. 84, No. 4, pp. 327-352, 1977.
16. C.J. van Rijsbergen, Information Retrieval, 2nd edition, Department of Computer Science, University of Glasgow, 1979.

On the Application of the Semantic Web Rule Language in the Definition of Policies for System Security Management

Félix J. García Clemente, Gregorio Martínez Pérez, Juan A. Botía Blaya, and Antonio F. Gómez Skarmeta

Departamento de Ingeniería de la Información y las Comunicaciones, University of Murcia, Spain
{fgarcia, gregorio, skarmeta}@dif.um.es, juanbot@um.es

Abstract. The adoption of a policy-based approach for the dynamic regulation of a system or service (e.g. security, QoS or mobility service) requires an appropriate policy representation and processing. In the context of the Semantic Web, the representation power of languages enriched with semantics (i.e. semantic languages), together with the availability of suitable interpreters, make such kind of languages well suited for policies representation. In this paper, we describe our proposal for the combination of the CIM-OWL ontology (i.e., the mapping of the DMTF Common Information Model into OWL) with the Semantic Web Rule Language as the basis for a semantically-rich security policy language that can be used to formally describe the desired security behaviour of a system or service. An example of security policy in this language and its reasonig are also presented.

1 Introduction

A policy-based management enables a system administrator to specify rules that describe domain-wide policies. These specifications are defined by using high level languages in such an extent that they are totally decoupled of any possible implementation of the system. There are multiple approaches for policy specification. Two examples are (1) formal policy languages that a computer can easily and directly process and interpret and (2) rule-based policy languages based on conventional if-then rules which includes the representation of policies based on a deontic logic for the expression of rules related to obligation and permissibility.

The adoption of a policy-based approach for security management requires an appropriate policy representation and an engine for policy processing that enables runtime adaptability and extensibility of the system, as well as the possibility of enabling analysis of policies relating to entities described at different levels of abstraction. In this sense, semantic approaches based on the combination of ontology languages [6] and rule-based languages satisfy these requirements. For example, organizations may utilize a common ontology that can be shared amongst services and service clients to define rule-based security policies. The use of ontologies also facilitates the process of reasoning over the structure and interrelations of a set of policies easing the detection of conflicts. This functionality is especially relevant in

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 69 – 78, 2005.
© Springer-Verlag Berlin Heidelberg 2005

medium and large organizations which may manage tens or hundreds of different policy rules at the same time.

One standard that provides a common model for describing the necessary concepts to be considered within a policy domain is the DMTF Common Information Model (CIM). The CIM specification is a semi-formal ontology that does not support interoperability and reasoning. These limitations are due, in part, to the constraints imposed by the languages in which the CIM model is specified (e.g. XML and XML Scheme). To overcome the mentioned drawbacks we may model, represent and share CIM in the form of a formal ontology. In this sense, the first part of the paper presents a CIM-OWL ontology (i.e., the specification of the DMTF Common Information Model in OWL, which stands for Ontology Web Language) in section 2. With this idea in mind, there is still a necessity of specifying the policy rules themselves. This is something that one may afford with OWL. However, OWL does not allow for specifying rules directly. Instead, one has to define rules skeleton as regular classes in the ontology. Clearly, this is not a suitable approach. For the policy reasoning part we have decided to adopt a specific language for defining and interpreting rules. There are multiples rule languages, and normally each one is designed and developed together with a concrete inference engine. One language that has been designed as a semantic interoperable vehicle for heterogeneous policy languages is Semantic Web Rule Language (SWRL). The SWRL language provides intermediate mark-up syntax, with associated deep knowledge representation semantics for interchange between those languages. For interchange between policy languages that are already XML-based, this may, for instance, be achieved using XSL transformations (XSLT), e.g., to translate into and then out of SWRL. This part of the work is described in section 3. The different implications of this approach are analyzed in section 4. Section 5 shows an example of a security policy, and how and what may be reasoned from it. Finally, section 6 outlines most important conclusions we have obtained and points out some new work directions we are now involved in.

2 A Summary of the CIM-OWL Ontology

CIM [2] is an approach from the DMTF that applies the basic modelling techniques of the object-oriented paradigm to provide a common definition of management-related information. It comprises a core model that defines a basic classification of elements and associations for a managed environment (e.g., logical and physical elements, capabilities, settings and profiles) as well as more specific models that define concepts that are common to particular management areas (e.g., applications, systems, devices and users).

CIM is independent of any implementation or specific specification. However, for an information model to be useful, it has to be mapped into some implementation. In this sense, CIM can be mapped to multiple structured specifications. For example, in [1] the authors describe a mapping from the CIM resource model and related operations to the Web services paradigm using XML and WSDL. This specification permits one to model the management of web services using the DMTF methodology and hence to obtain its standard representation. Whereas the XML-encoded CIM specification can not be arbitrarily combined with other specifications in a flexible manner and, with respect to more advanced operations, XML-encoded specifications

do not embody the constructs for facilitating tasks like parsing, logical deduction or semantic interpretation.

Other specific specification of CIM may be performed by using OWL [7]. This representation of CIM makes easy to perform useful reasoning tasks. This is due to the fact that OWL is based on description logics [8]. This simple and yet powerful kind of first order like logic allows for reasoning not only on individuals but also on the structure of them with efficient and sound algorithms. In this sense, we presented a proposal for expressing CIM objects in OWL, named CIM-OWL, in [3]. We have now extended this proposal with the mapping of all CIM qualifiers (i.e. meta, standard and optional) to OWL, as it is neccesary to preserve all information appearing at the original model. Table 1 shows the proposed mapping of CIM qualifiers to OWL.

Table 1. CIM qualifiers to OWL

Qualifier	*How it can be mapped*
Meta Qualifier	
Association	Indicates that the object class is defining an association. The association class will include the <owl:ObjectProperty> tag, while any other type of classes will not.
Indication	Indicates that the object class is defining an indication. It is mapped to this RDF scheme feature: <rdfs:subClassOf rdf:resource="#Indication" />
Standard Qualifier	
Abstract	Indicates that the class is abstract and serves only as a base for new classes; it is mapped to this RDF scheme feature: <rdfs:subClassOf rdf:resource="#Abstract" />
ArrayType	Indicates the type of the qualified array; it is mapped to <rdf:type rdf:resource="#ArrayType" />
Deprecated	Indication that the entity is deprecated; it is related with a versioning feature: <owl:deprecatedClass> or <owl:deprecatedProperty>
Description	Provides a description of a property, an operation or a class; it is mapped to <rdfs:comment>
Optional Qualifier	
Alias	Establishes an alternate name for a property; it results in an equality feature: <owl:equivalentProperty>
Invisible	Indicates that the element is defined only for internal purposes; it is mapped to this RDF scheme feature: <rdfs:subClassOf rdf:resource="#Invisible" />

The main rules for this mapping are the following:

- If a qualifier has an equivalent representation in OWL, it is used; it is the case of the qualifiers Deprecated or Description, for example.
- If a class qualifier does not have an equivalent representation in OWL, then we propose a new class to represent it, as with the qualifiers Indication or Abstract, for example.
- If a property qualifier does not have an equivalent representation in OWL, then we propose a new class that represents a new property which model the qualifier. It is the case of the qualifier ArrayType, for example.

The automatic transformation between the XML and OWL representations of CIM can be made by defining XSL templates implementing the indicated transformations.

3 Specification of Security Policies with Semantic and Rule Oriented Languages

Semantic Web Rule Language (SWRL) [4] is based on a combination of the OWL DL Lite language of the OWL Web Ontology Language family with the Unary/Binary Datalog RuleML sublanguages. SWRL extends the set of OWL axioms to include a high-level abstract syntax for Horn-like rules that can be combined with an OWL knowledge base. In this manner, OWL is used to define the pieces of knowledge appearing at the logic expressions within the body and head of rules and RuleML is used to define the reasoning procedure over that knowledge.

The SWRL rules are of the form of an implication between an antecedent (body) and consequent (head). The intended meaning can be read as: whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold.

A useful restriction in the form of the rules is to limit antecedent and consequent atoms to be named classes, where the classes are defined purely in OWL. Adhering to this format makes it easier to translate rules to or from future or existing rule systems, including Prolog, and Jena [5]. In the case of Jena, version 2.0 was used with the two rule systems available: forward and backward chaining, being the rules automatically transformed from SWRL version 0.5 to Jena 2.0.

Since CIM permits to represent both high-level and low-level concepts (e.g. application-level data versus network-level end points), we can use OWL-CIM plus SWRL to represent both high-level and low-level policies (e.g. “the integrity of all application data must be guaranteed” versus “block UDP connections to port 53 on host X”). Thus, policy administrators have enough flexibility to choose the appropriate level for their needs.

The specification of policies is performed in two different phases. The first one consists on selecting the correct OWL-CIM concepts to model the real system components and, if necessary, network topology. The second one deals with using the previously defined concepts to specify policy rules and, subsequently, the policies.

4 Automated Reasoning on Security Policies

The combination of the CIM-OWL ontology and SWRL for specifying behaviour rules for policies offers a clear advantage: it allows two different types of automated reasoning. The first one is ontology reasoning (i.e. reasoning over the structure and instances of the ontology) and the second one is rule-based reasoning (i.e. applying policy rules in systems management tasks). Therefore, we identify an OWL reasoner and a rule-based reasoner, where we use the term reasoner to refer to a specific program that performs the task of inference (i.e. process of deriving additional information no explicitly specified).

In the first phase of policy specification, the OWL reasoner may be used for:

- *Validation*. The OWL ontology language allows constraints to be expressed; the validation operation is used to detect when such constraints are violated by some data

set, i.e., the validation consists on a global check across the schema and instance data looking for inconsistencies.

- *Query the model for* instance recognition (i.e., testing if an individual is an instance of a class expression) and inheritance recognition (i.e., testing if a class is a subclass of another class and if a property is a sub-property of another).

In the second phase of policy specification, the rule reasoner is used to obtain additional knowledge not explicitly specified from the system definition. It provides forward and backward chaining reasoning and while the OWL reasoner performs inference about OWL-CIM ontology, the rule reasoner does about SWRL rules.

Moreover, the rule reasoner eases the detection of conflicts. A conflict occurs when the policy definition assigns different specifications on the behaviour of system component, e.g., one allows the user to start the service and another prohibits the same user from starting the service. It may be used to detect both static and dynamic conflicts. Static conflict detection aims to detect all types of potential conflicts (possible or definite) which clearly could cause conflicts from the policy specification. This static conflict detection is performed on the process of policy definition. Unlike static conflict detection, dynamic conflict detection is performed at run time by dynamically detecting all conflicts whenever the system definition is modified.

5 An Example

This section shows both ontology reasoning and rule-based reasoning examples over a particular authorization policy defined from a portion of the CIM-OWL ontology.

5.1 The Ontology for the Example

This example shows the subset of CIM classes necessary to express authorization policies between computer systems and roles. We use the CIM classes depicted in Figure 1 to represent the management-related concepts regarding the authorization security service, computer systems, and roles.

The *Privilege* object class is the base for all types of activities, which are granted or denied to a subject by a target. *AuthorizedPrivilege* is the specific subclass for the authorization activity. Whether an individual Privilege is granted or denied is defined using the *PrivilegeGranted* boolean. The association of subjects to *AuhorizedPrivileges* is accomplished explicitly via the association *AuthorizedSubject*. The entities that are protected (targets) can be similarly defined via the association *AuthorizedTarget*. Note that *AuthorizedPrivilege* and its *AuthorizedSubject/Target* associations provide a static mechanism to represent authorization policies.

The *Role* object class is used to represent a position or set of responsibilities within an organization, organizational unit or system administration scope. It is filled by a person or persons (or non-human entities represented by *ManagedSystemElement* subclasses) that may be explicitly or implicitly members of this collection subclass.

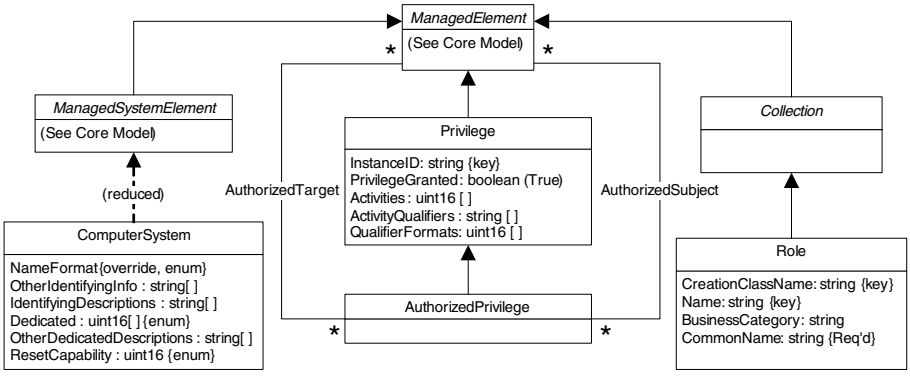


Fig. 1. UML diagram of CIM classes

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
 xmlns:owl="http://www.w3.org/2002/07/owl#"
 xml:base="http://www.positif.com/cim#"
 xmlns:cim="http://www.positif.com/cim#"
 xmlns="http://www.positif.com/cim#">
<CIM_ComputerSystem rdf:about="#printserver">
        <Caption>Print Server</Caption>
        <ElementName>Print Server</ElementName>
        <Name>printserver.positif.org </Name>
        <Dedicated>Print</Dedicated>
</CIM_ComputerSystem>
<CIM_AuthorizedTarget rdf:about="#authtarget1">
        <CATPrivilege rdf:resource="#printauth" />
        <TargetElement rdf:resource="#printserver" />
</CIM_AuthorizedTarget>
<CIM_AuthorizedPrivilege rdf:about="#printauth">
        <Caption>Access Authorization for the print server</Caption>
        <ElementName>Print Authorization</ElementName>
        <InstanceID>POSITIF:PrintAuth</InstanceID>
        <PrivilegeGranted>true</PrivilegeGranted>
        <Activities>Create</Activities>
</CIM_AuthorizedPrivilege>
<CIM_AuthorizedSubject rdf:about="#authsubject1">
        <CASPrivilege rdf:resource="#printauth" />
        <PrivilegedElement rdf:resource="#ST_PHY" />
</CIM_AuthorizedSubject>
<CIM_Role rdf:about="#ST_PHY">
        <Caption>Students of Philosophy</Caption>
        <ElementName>PhilosophyStudents</ElementName>
        <CreationClassName>CIM_Role</CreationClassName>
        <Name>ST_PHY</Name>
        <BusinessCategory>Students</BusinessCategory>
</CIM_Role>
<CIM_Role rdf:about="#ST_COMP">
        <Caption>Students of Computer Science</Caption>
        <ElementName>ComputerStudents</ElementName>
        <CreationClassName>CIM_Role</CreationClassName>
        <Name>ST_COMP</Name>
        <BusinessCategory>Students</BusinessCategory>
</CIM_Role>
</rdf:RDF>
```

Fig. 2. Representation in OWL-CIM of the administrative domain

The *ComputerSystem* object class is derived from *System* that is a special collection of *ManagedSystemElements* that provides computing capabilities. The *Dedicated* property is an enumeration indicating whether the *ComputerSystem* is a special-purpose System (i.e., dedicated to a particular use), versus being general purpose. For example, one could specify that the System is dedicated to "Print" (value=11) or acts as a "Hub" (value=8).

Since both *ComputerSystem* and *Role* are specializations of *MagamentElement*, they may be either the target or the subject of *Privilege*. Our example uses *Role* as subject and *ComputerSystem* as target.

In this particular example, the administrative domain is composed by a system dedicated to print, a role that represent the set of students of Computer Science, and a role that represent the set of students of Philosophy. It occurs that students of Philosophy have the privilege granted to print within the system, whereas students of Computer Science do not have it. Figure 2 shows the OWL-CIM representation of the administrative domain.

5.2 Reasoning over the Ontology

For this administrative domain, the system administrator may validate the system definition and find inconsistencies by using the inference engine. For example, Figure 3 shows a OWL definition with one inconsistence. The inference engine infers that the *CATPrivilege* property references a resource with a non expected type, since the resource *ST_PHY* is a *Role*, and it must be an *AuthorizedPrivilege*. Other inconsistencies that the OWL reasoner can detect are, for example, property cardinality and data types.

The policy administrator may also test the definition by querying about classes and instances. For example, he/she may perform the following question: *"Is PrintServer a ManagedSystemElement?"* The OWL reasoner recognizes the instance as ManagementElement. More questions that the OWL reasoner may answer are related to CIM qualifiers, e.g., "Is ManagedSystemElemet abstract?"

```
<CIM_AuthorizedTarget rdf:about="#authtarget1">
        <CATPrivilege rdf:resource="#ST_PHY" />
        <TargetElement rdf:resource="#printserver" />
</CIM_AuthorizedTarget>
```

Fig. 3. Representation in OWL-CIM with one inconsistence

5.3 Reasoning with Policy Rules

For this administrative domain, the policy administrator may decide the following authorization policy: *"If a system permits to print to the students of Philosophy, then the students of Computer Science can also use this system to print"*. Figure 4 shows the SWRL representation of this authorization policy.

```
<ruleml:imp>
<ruleml:_rlab ruleml:href="#exampleRule"/>
<ruleml:_body>
   <swrlx:classAtom>
      <owlx:Class owlx:name="#CIM_AuthorizedTarget" />
      <ruleml:var>authtarget</ruleml:var>
   </swrlx:classAtom>
   <swrlx:classAtom>
      <owlx:Class owlx:name="#CIM_AuthorizedSubject" />
        <ruleml:var>authsubject1</ruleml:var>
   </swrlx:classAtom>
   <swrlx:classAtom>
      <owlx:Class owlx:name="#CIM_ComputerSystem" />
      <ruleml:var>printer</ruleml:var>
   </swrlx:classAtom>
   <swrlx:classAtom>
      <owlx:Class owlx:name="#CIM_AuthorizedPrivilege" />
      <ruleml:var>privilege</ruleml:var>
   </swrlx:classAtom>
   <swrlx:datavaluedPropertyAtom swrlx:property="#Dedicated">
      <ruleml:var>printer</ruleml:var>
        <owlx:DataValue
             owlx:datatype="&xsd;string">Print</owlx:DataValue>
   </swrlx:datavaluedPropertyAtom>
   <swrlx:individualPropertyAtom swrlx:property="#CATPrivilege">
      <ruleml:var>authtarget</ruleml:var>
      <ruleml:var>privilege</ruleml:var>
   </swrlx:individualPropertyAtom>
   <swrlx:individualPropertyAtom swrlx:property="#TargetElement">
      <ruleml:var>authtarget</ruleml:var>
      <ruleml:var>printer</ruleml:var>
   </swrlx:individualPropertyAtom>
   <swrlx:individualPropertyAtom swrlx:property="#CASPrivilege">
      <ruleml:var>authsubject1</ruleml:var>
      <ruleml:var>privilege</ruleml:var>
   </swrlx:individualPropertyAtom>
   <swrlx:individualPropertyAtom swrlx:property="#PrivilegedElement">
      <ruleml:var>authsubject1</ruleml:var>
      <owlx:Individual owlx:name="#ST_PHY" />
   </swrlx:individualPropertyAtom>
</ruleml:_body>
<ruleml:_head>
   <swrlx:classAtom>
      <owlx:Class owlx:name="#CIM_AuthorizedSubject" />
      <ruleml:var>authsubject2</ruleml:var>
   </swrlx:classAtom>
   <swrlx:individualPropertyAtom swrlx:property="#CASPrivilege">
      <ruleml:var>authsubject2</ruleml:var>
      <ruleml:var>privilege</ruleml:var>
   </swrlx:individualPropertyAtom>
   <swrlx:individualPropertyAtom swrlx:property="#PrivilegedElement">
      <ruleml:var>authsubject2</ruleml:var>
      <owlx:Individual owlx:name="#ST_COMP" />
   </swrlx:individualPropertyAtom>
</ruleml:_head>
</ruleml:imp>
```

Fig. 4. SWRL representation of an authorization policy

This SWRL rule together with the ontology can be load into a rule reasoner, for example into the Jena reasoner (the one used during this research work). Figure 5 shows the Jena representation of this authorization policy. Other reasoners such as Pellet or CLIPS can also be used.

```
#exampleRule:
( ?authtarget  http://www.w3.org/1999/02/22-rdf-syntax-ns#type
       http://www.positif.com/cim#CIM_AuthorizedTarget )
( ?authsubject1  http://www.w3.org/1999/02/22-rdf-syntax-ns#type
       http://www.positif.com/cim#CIM_AuthorizedSubject )
( ?printer  http://www.w3.org/1999/02/22-rdf-syntax-ns#type
       http://www.positif.com/cim#CIM_ComputerSystem )
( ?privilege  http://www.w3.org/1999/02/22-rdf-syntax-ns#type
       http://www.positif.com/cim#CIM_AuthorizedPrivilege )

( ?printer http://www.positif.com/cim#Dedicated 'Print' )
( ?authtarget http://www.positif.com/cim#CATPrivilege ?privilege )
( ?authtarget http://www.positif.com/cim#TargetElement ?printer )
( ?authsubject1 http://www.positif.com/cim#CASPrivilege ?privilege )
( ?authsubject1 http://www.positif.com/cim#PrivilegedElement
       http://www.positif.com/cim#ST_PHY )
->
( ?authsubject2  http://www.w3.org/1999/02/22-rdf-syntax-ns#type
       http://www.positif.com/cim#CIM_AuthorizedSubject )
( ?authsubject2 http://www.positif.com/cim#CASPrivilege ?privilege )
( ?authsubject2 http://www.positif.com/cim#PrivilegedElement
       http://www.positif.com/cim#ST_COMP )
]
```

Fig. 5. Jena representation of an authorization policy

The rule reasoner infers new data that grants the privilege to print by the system to the students of Computer Science. Figure 6 shows the representation OWL-CIM of the data inferred by rule reasoner.

```
<CIM_AuthorizedSubject rdf:about="#authsubject2">
       <CASPrivilege rdf:resource="#printauth" />
       <PrivilegedElement rdf:resource="#ST_COMP" />
</CIM_AuthorizedSubject>
```

Fig. 6. Data inferred by rule reasoner

6 Conclusions and Future Work

This paper describes a semantic approach for the representation of security policies based in OWL and SWRL. It emphasizes two main ideas. The first one is that, due to the use of OWL it is possible to perform some reasoning over the knowledge model. The second one is that, due to the use of SWRL, it is possible to perform also reasoning over the rules to correctly handle system management policies. We believe that this approach for system management is a strong bid for the future of distributed and complex systems like, for example, Web services based systems, multi-agent systems and peer to peer systems.

Currently, we have a preliminary version of the OWL reasoner and rule-based reasoner for security policies represented by the combination of OWL-CIM ontology and SWRL. Our deployment is based in the use of Jena 2 inference subsystem. The CIM model has still work to be done on the completion of the whole model as it is very complex in terms of the number of modelling entities involved. We are also developing a graphical editor for policies. This represents ontologies as hierarchies of elements, and rules may be easily defined by just selecting elements on the ontology

to be part either of the body or of the head or each rule. This is very convenient as the operator (i.e., the person which is in charge of defining and administrating policies) does not need to be familiar with technological aspects like ontologies and automated reasoning. He/she only needs to know about the knowledge model shown by the ontology. Moreover, we are also working to determine if the power of SWRL is enough to capture all the policy concerns.

Acknowledgements

The work presented in this article has been partially funded by the EU in the context of the POSITIF (Policy-based Security Tools and Framework) IST project (IST-2002-002314) and also by the ENCUENTRO (00511/PI/04) Spanish Seneca project.

References

1. García, F. J, G. Martínez, O. Cánovas, and A.F. Gómez-Skarmeta: A Proposal of a CIM-Based Policy Management Model for the OGSA Security Architecture, GADA Workshop, OTM Workshops 2004, 10/2004.
2. Common Information Model (CIM) Standards, DMTF, http://www.dmtf.org/standards/cim, WWW, 2005.
3. García, F. J., G. Martínez and A. F. Gómez-Skarmeta, A Semantically-Rich Management System based on CIM for the OGSA Security Services, In Knowledge and Data Mining Grid Workshop, 3rd Atlantic Web Intelligence Conference, 6/2005.
4. SWRL: A Semantic Web Rule Language Combining OWL and RuleML, The Rule Markup Initiative, http://www.ruleml.org/swrl/, WWW, 2005.
5. Jena – A Semantic Web Framework for Java, http://jena.sourceforge.net/, WWW, 2005
6. Fensel, D. Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce. Springer-Verlag, 2004.
7. Smith, M.K., and C. Welty and D.L. McGuinness. OWL Web Ontology Language Guide. W3C Recommendation. W3C, 2004.
8. Baader F., D. Calvanese , D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider. An Introduction to Description Logics, In Description Logic Handbook. Cambridge University Press, 2002.

On Secure Framework for Web Services in Untrusted Environment

Sylvia Encheva[1] and Sharil Tumin[2]

[1] Stord/Haugesund University College, Bjørnsonsg. 45, 5528 Haugesund, Norway
sbe@hsh.no
[2] University of Bergen, IT-Dept., P. O. Box 7800, 5020 Bergen, Norway
edpst@it.uib.no

Abstract. In this paper we identify trust relationships among users and systems. We try to adhere to simplicity principle in our modelling of the system. By using simple model and free lightweight technologies, we show that it is possible to implement secure Web applications/services. The paper also addresses some security problems and issues about implementing Web Services.

1 Introduction

Security within information systems is seen differently from different prospectives. To a user, security usually means a protection of privacy and identity theft and a protection against framing. To a system, security usually means protection of data and process integrity, information flow and resources. The {user, system}-pair leads to a necessary establishment of the following trust relations; `user-system, system-user, user-user`, and `system-system`. In practice these trust relations are made mutual by, 'I trust you if you trust me' principle. For example, a system trusts a user if the user provides a valid credential at sign-on, a user in turn trusts a system to protect both data and processes such that, user's identity is not being compromised. Whose fault is it when an identity is caught doing an illegal act? Is it a dishonest user, who is the owner of the identity, or a security weak system, that allows identities theft to occur? It might very well be the fault of a weak communication link protocol that leaks users' identities under the establishment of the above mentioned trust relations.

Web services and Web-based applications promise mobile users assessable services from anywhere at any time. In the last decade one can see many evolving technologies and standards supporting secure framework for Web services/ applications. The Web is based on a model of request/response connectionless mode of communication relaying on the stateless HyperText Tranfer Protocol (HTTP) communication protocol. Its simplicity is the major factor for its widespread success.

Lately Web technologies have evolved to a high degree of sophistication employing complicated protocols supported by heavy-weight software Java based systems. Among these are XML, SOAP, WDSL, UDDI and SAML endorsed by multinational software giants like Microsoft, IBM, Oracle and Sun.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 79–88, 2005.
© Springer-Verlag Berlin Heidelberg 2005

In this paper we propose a secure framework based on open-source softwares for Web applications/services for small teams of developers and implementers on low budget development projects, common to educational organizations. Small means not more then four in a team and low budget means no software and license costs. Web applications/services have been developed and deployed due to necessity and not based on commercial goals. Developers and engineers, members of such development teams, normally have different levels of technical knowledge, experience and know-how. Usually, such a project concentrates on workability of a system in a complex environment rather than producing a commercial grade software for an assumed environment. To meet the workability goal, security concerns are not taken into consideration due to lack of experience and/or work knowledge. We believe that by using simple and open-ended software tools, developers, and implementers can achieve both workability and a higher level of security due to the fact that a system being developed is under a full control of the developers. Security concerns must be a big part of the implementation model and must not be taken as an afterthought once the system is put in production.

The paper is organized as follows. Open-source software tools used in this framework are presented in Section 3. Trust relations are discussed in Section 4. In Section 5 we use XML-RPC for Peer-to-peer (P2P) communication that ensures security, privacy and non-repudiation. A method of using password card called PASS-card for Web sign-on that does not disclose users' system credentials is presented in Section 6. The paper ends with a conclusion in Section 7.

2 Related Work

Network security problems are discussed in [1]. A set of hints for designing a secure client authentication scheme is described in [7]. A taxonomy of single sign-on systems is presented in [10].

Pubcookie [4] provides an open-source software for intra-institutional single sign-on (SSO) end-user Web authentication.

PGPi [5] is the international variant of Pretty Good Privacy (PGP), that provides an email encryption system. PGP is normally used to apply digital signatures to emails and it can also encrypts emails, and thus provides privacy.

PGP does not depend on the traditional hierarchical trust architecture but rather adopts the 'web of trust' approach [11]. Trust issues related to network are discussed in [9].

Limitations to existing e-commerce technologies: data resides in traditional databases, and security is difficult to guarantee across network [2].

Practical sides of Public Key Infrastructure (PKI) are presented in [3].

3 Software Tools

Python is chosen primarily for its simplicity and expressiveness and secondarily for the many open-source modules developed by the Internet communities that

support Web application/services and security tools. We choose the XML-RPC, remote procedure call encapsulated in Extensible Markup Language (XML) for P2P communication for its simplicity in implementation and usage, rather then more complicated technologies like Simple Object Access Protocol (SOAP). By employing secure library from OpenSSL and Python crypto modules such as CryptoTools and M2Crypto we manage to do P2P securely.

Apache is a robust and extendable HTTP server. It is one of the most stable and secure services that ships with many Linux/Unix distributions. By using options under installation and a flexible configuration file for run-time, the Apache HTTP can be made secure. By an easy and understandable change of configuration, it can be made to support a secure protocol (HTTPS). Web resources can be secured using well understood authentication module that involves password and client certificates. In this paper we investigate the possibility of securing Web application/services without using HTTPS and certificates.

PostgreSQL, with more than fifteen years of development history, is chosen for its scalability, Structured Query Language (SQL) compliant and object-relational database. PostgreSQL supports Procedural Language (PL) in different programming languages like PL/pgSQL, PL/Perl, PL/Tcl, and PL/Python. Thus PostgreSQL functions can be written in Python. PostgreSQL supports several methods of authentication and authorization to a database by controlling clients (client-IP address) and users (credential-user ID and password) access. Clients can be 'local', 'hosts', and 'hostssl'. A user can be completely trusted by 'trust' or completely blocked by 'reject'. A user can transfer his/her credentials by 'password' (clear text), 'crypt' (encrypted password), krb4, krb5 (Kerberos) and 'ident' (identification protocol-RC1413). All these are done in a `pg_hba.conf` file which is read every time a {client-user} pair needs to be authenticated. The database server needs not be restarted to make a new security policy in `pg_hba.conf` effective.

4 User-System, System-User, User-User and System-System Relations

The `user-system` trust relation concern is about the question of users' anonymity and privacy. The users trust the system to protect their personal data. Users' credentials and authorization data are protected by a secure user-administration system. Users' data must not leak out to others neither intentionally nor by mistake. When a user gives his/her credentials or other sensitive information to a system, he/she needs to be sure that these data really go to the intended server.

The `system-user` trust relation concern is about users' authentication and authorization. The system trusts the credentials provided by a user. Users are responsible for protecting their credentials. The system must provide users with strong password policies, and a framework where users' credentials will not be compromised by a weak security implementation. An SSO that maps a single action of user authentication to many access permissions provides users with the

convenience of using only one credential globally within a collection of systems. A user needs not be given multiple identification-password pairs and once authenticated, needs not be asked to be authenticated again. SSO reduces human error, that appears to be a major component of security failure, and is therefore highly desirable. However, a stolen credential opens the possibility of a security breakdown globally. The need to protect users' credentials is of a paramount importance.

The `user-user` trust relation concern is about communicating party's authenticity between them. The sender wants to make sure that only the receiver can read a sent message. The receiver wants to be sure that the message received really originated from the sender. Once a message is received by the receiver, the sender cannot refute that the message did not come form him/her. The sender and the receiver agree on a non-refutable mutual contract on the originality and validity of the messages passed between them.

The `system-system` trust relation is similar in context to the `user-user` trust relation. Here the sender and receiver are communicating programs across an insecure channel. A message can be a remote procedure request or response. The message needs to be protected against disclosure and tempering a route.

Systems within an organization or federated organizations can build and manage a PKI. The PKI assumes the use of public key cryptography to enable users of a basically insecure public network such as the Internet to securely and privately exchange data. Users' public keys are made publicly accessible via Lightweight Directory Access Protocol (LDAP) directory services.

However, not all security concerns can be programmed. A secure system must be supported by site's rules and regulations, controlled by a body of a security committee. Such a committee must be lead by a high-level official. The committee provides security guidelines, responsible for security auditing and has the power to apply sanctions.

5 Signed Digital Envelope

In our framework an XML message (*payload*) is encrypted by a symmetric cryptographic function (*crypto_func*) using a secret-key (*skey*). Public-key cryptography is used to encrypt the secret key using the public-key of the receiving party. Together, they make a message in a digital envelope.

The sender takes the digital envelope and runs it through a hash function (*hash_func*) to produce a hash value (Fig. 1). A one-way hash function generates a unique text string to the given input. The hash value is then encrypted using the sender's private key to create a digital signature (*signature*) and this authenticates the sender, since only the owner of that private key could encrypt the message.

The *crypto_func, skey, hush_func, signature* and *payload* are then used as parameters to an XML-RPC request and return values to the call. The actual procedure name and its parameters or the actual return values are embedded in the *payload*.

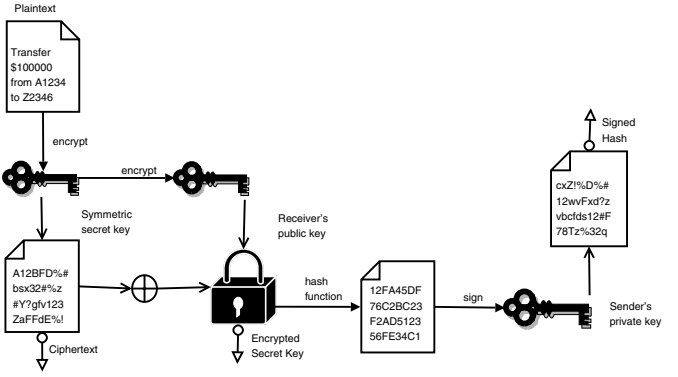


Fig. 1. Sender process

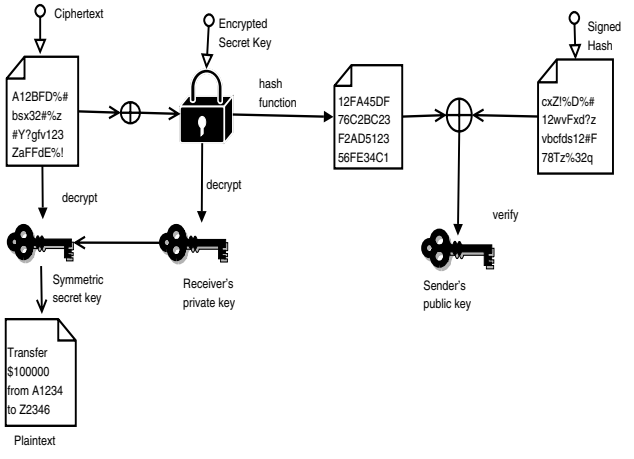


Fig. 2. Receiver process

On message arrival, the receiver unpacks the XML message and does the reverse process of decryption and verification (Fig. 2).

For an XML-RPC request, the receiver unpacks the *payload* to get the procedure name and its parameters. On XML-RPC response, the receiver unpacks the *payload* to get return values. Actually, the payload data is a datastructure made into XML by using a Python's xmlrpclib module.

The *skeys* used are made different for different messages. The requester signs its request message and the responder signs its response message. Thereby we manage to use XML-RPC for P2P communication that ensures security, privacy and non-repudiation.

6 SSO

The use of a single credential {user-identification, password}-pair for a system wide authentication provides users with the convenience of remembering one

password only. The same {user-identification, password}-pair is used to logon into Unix, Windows and other servers. However, a single credential policy increases the risk of the system wide security breach, should that credential got stolen. It is especially risky when the single credential is used for Web sign-on from a Web browser situated in non-trusted environments like Web Cafe, public libraries and the like. A keyboard grabber program can easily steal users' credentials without users knowledge. One solution is not to use a {user-identification, password}-pair credentials for Web applications' sign-on. Some of the technologies supporting such a solution are Smart-cart, biometric devices, and a {client certificate, pin}-pair method. Normally such devices are non-existing under the environments mention above. Users may not be able or may not want to install a client certificate on publicly accessible PCs. We propose a method of using password card called PASS-card for Web sign-on that does not disclose users' system credentials.

A user can produce such a card (a randomly generated image) via a Web application from a PC within a trusted network, like f. ex. organization's internal network, at anytime.

6.1 Producing a PASS-card

A user can produce a PASS-card (a randomly generated image) via a Web application from a PC within a trusted network, like for example organization's internal network, at anytime. A user has to choose a nickname and a PIN-code while producing a PASS-card. A PASS-card contains twelve couples and a serial number (Fig. 3). Each couple consists of two randomly generated characters.

During any process of sign-on, a user is asked to provide a nick-name connected to his/her PASS-card. The sign-on application then randomly picks three out of the twelve couples on the PASS-card. These couples form a PASS key for this particular use of the card.

The sign-on application asks the user to provide a PASS key (Fig. 4). The PASS key contains three pairs: the first pair (12) (Fig. 4) which corresponds to the couple *jd* (Fig. 3); the second pair (34) (Fig. 4) which corresponds to the couple *eG* (Fig. 3) and the third pair (56) (Fig. 4) which corresponds to the couple *ih* (Fig. 3). The resulting sequence 'jdeGih' is the user's PASS key

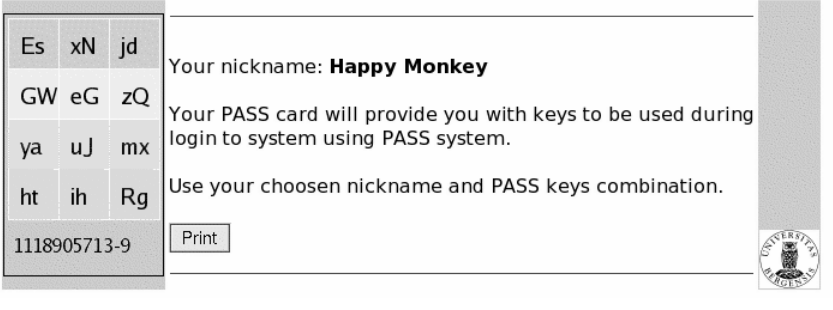


Fig. 3. Pass card request

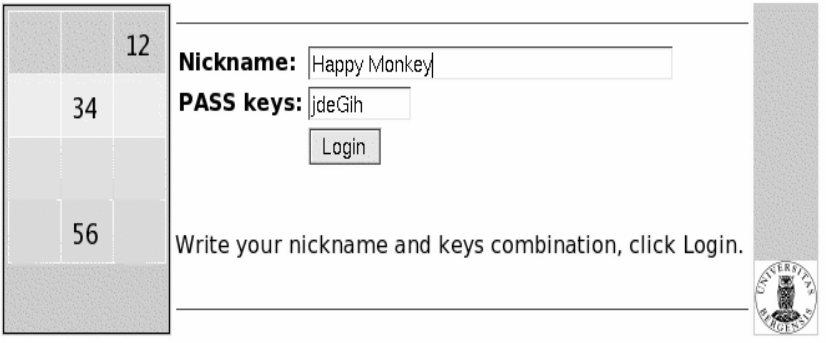


Fig. 4. Pass card sign-on

(Fig. 4). The PASS-card map on the left side on Fig. 4 is an image dynamically created by the system using the Python's GD module.

6.2 PASS-card Coordinates and a PASS-card Key

The user proves his/her authenticity to the system by a correct mapping of the PASS-card coordinates to a PASS-card key, and by providing a valid nickname. If the user succeeds, the system then proves its validity by presenting the user with the PASS-card serial number. The sign-on is successful if the user can then provide a valid PIN-code. If the triplet {Nickname, PASS keys, PIN-code} is valid then it is mapped to the real system user. A user can revoke his/her PASS-card from anywhere and obtain a new one within a trusted network at anytime.

6.3 System Framework

We now discuss the system framework for secure SSO mechanism without the need of using HTTPS and certificates. The framework has the following stages:

`Stage 1`, described in steps 1, 2, and 3 (Fig. 5);
`Stage 2`, described in steps 4, 5, and 6 (Fig. 6);
`Stage 3`, described in steps 7, 8, and 9 (Fig. 7).

Assume that a user is not yet authenticated.

`Stage 1`

1. A user is accessing a protected Web application. If there is no valid session cookie the AP server shows an initial sign-on Web form (Fig. 5). The user provides a domain sign-on server name (SO server). If the user's browser has a valid session cookie then the AP server gives the user an access.
2. The AP server connects to the SO server (XML-RPC without signed digital envelope support), sends its public-key (Pp), and receives SO server's public-key (Sp) in reply. Public-keys are saved in Pub_Keys.

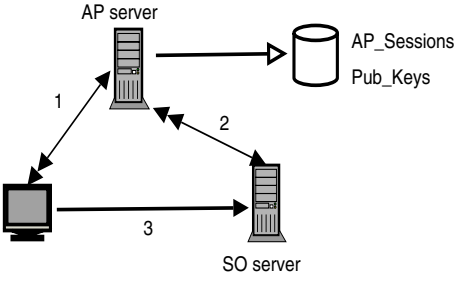


Fig. 5. Stage 1

3. The AP server creates a new session cookie (*Cn*) and saves *Cn* and the SO server's address in AP_Sessions. The AP server sets *Cn* in user's browser and redirects the user's browser with the AP server's name and *Cn* as HTTP-GET parameters to the SO server for domain sign-on.

Stage 2

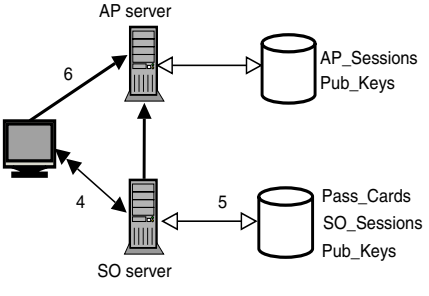


Fig. 6. Stage 2

4. If there is no valid session cookie the SO server creates a new session cookie (*Cs*) and saves it (Fig. 6). The SO server sets *Cs* in user's browser and shows a Web form for domain sign-on. The SO server first asks the user to provide a sign-on identifier (*SID*). The SO server matches the *SID* with the user's PASS-card in the database. It then constructs a random PASS-card image (Fig. 4) for the user's sign-on Web form. The user's *SID* is different from the domain user identifier (*UID*). The user is authenticated, if the user provides the correct token in relation to the given PASS-card image. The SO server saves the authenticated user's *UID* to the *Cs*.
5. The SO server connects to the AP server (XML-RPC with signed digital envelope support) and sends *Cn* (created under step 3) and *Cs*. The AP server verifies and unpacks the data package, and then saves *Cs* to *Cn*.
6. The SO server redirects the user's browser back to the AP server. At this moment the user's browser contains *Cs*→ *UID* for global sign-on and sign-off control, and *Cs*→*UID* session user binding to protected application.

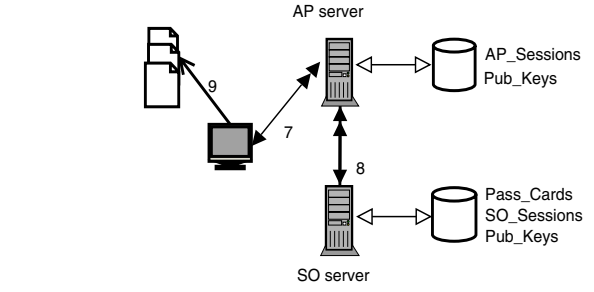


Fig. 7. Stage 3

7. The AP server first reads the user's browser cookie *Cn*, and then consults AP_Session for *Cs* for the given *Cn* (Fig. 7). The *Cs* at the SO server points to the authenticated user *UID*.
8. The AP server connects to the SO server (XML-RPC with signed digital envelope support) and sends *Cs* (from step 7). The SO server verifies and unpacks the data package, and then consults SO_Sessions for *UID* with the given *Cs* and *Cn*. The SO server responds with an *UID*.
9. The AP server creates a new user session cookie (*Cu*) for *UID*. The AP server sets *Cu* in the user's browser and redirects the user's browser to protected Web resources using *Cu* as a session validation.

7 Conclusion

PASS-cards provide users to sign-on from virtually anywhere (by using only http) without fear of disclosing their real system credentials. The users themselves administer the usage and validity of the PASS-cards they own. XML-RPC with signed digital envelope makes it possible to transmit request/response messages trustworthily, securely and privately over an insecure public network.

One of the major problems of security framework has to do with key distribution and management. We assume that an organization maintains a domain wide user administration system. Domain users are provided with a Web application from which they can create and revoke public-private key pairs. A Web application is also provided by the domain server for signing and verifying users' messages. Domain users public keys are published in an LDAP directory server. Collaborating organizations within the framework implicitly trust each other domain users' public keys, published in the domains' LDAP servers.

References

1. Albanese, J, Sonnenreich, W.: Network Security Illustrated, McGraw-Hill Professional, (2003)
2. Garfinkel, S.: Web Security, Privacy & Commerce, O'Reilly, (2002)

3. Geschwinde E., Schönig H-J.: PostgreSQL, Developer's Hadbook, Sams Publishing, USA (2001)
4. http://www.pubcookie.org
5. http://www.pgpi.org
6. Ferguson, N., Schneier B.: Practical Cryptography, Wiley, (2003)
7. Fu, K., Sit, E., Smith, K., Feamster, N.: Dos and Don'ts of Client Authentication on the Web. Proceedings of the 10th USENIX Security Symposium, Washington, D.C., August, (2001)
8. Herzberg A., Mass Y., Mihaeli J., Naor, D., Ravid, Y.: Access control meets public key infrastructure, or: Assigning roles to strangers. EIRE Symposium on security and privacy, (2000)
9. Lu, Y., Wang W., Xu, D., Bhargava, B.: Trust-based Privacy Preservation for Peer-to-peer Data Sharing. Proceedings of the 1st NSF/NSA/AFRL workshop on Secure Knowledge Management (SKM), (2004)
10. Pashalidis, A., Mitchell, C.J.: A taxonomy of single sign-on systems. Lecture Notes in Computer Science 2727, 249-264, (2003)
11. Zimmermann, P.: Pretty Good Privacy User's Guide, Distributed with the PGP software, (1993)

An Approach for Semantic Query Processing with UDDI

Jim Luo, Bruce Montrose, and Myong Kang

Center for High Assurance Computer Systems, Naval Research Laboratory,
Washington, DC 20375
{luo, montrose, mkang}@itd.nrl.navy.mil

Abstract. UDDI is not suitable for handling semantic markups for Web services due to its flat data model and limited search capabilities. In this paper, we introduce an approach to allow for support of semantic service descriptions and queries using registries that conforms to UDDI V3 specification. Specifically, we discuss how to store complex semantic markups in the UDDI data model and use that information to perform semantic query processing. Our approach does not require any modification to the existing UDDI registries. The add-on modules reside only on clients who wish to take advantage of semantic capabilities. This approach is completely backward compatible and can integrate seamlessly into existing infrastructure.

1 Introduction

Automatic discovery of Web services is an important capability for the Service-Oriented Architecture (SOA). The first step in providing this capability is to mark up Web services with metadata in a well-understood and consistent manner. The W3C community developed the Web ontology language (OWL) to address this problem [1]. It is a machine understandable description language that is capable of describing resources in a manner much richer than the traditional flat taxonomies and classification systems. OWL-S is a set of ontology developed specifically to describe web services [2]. After the semantic service descriptions are created, the next step is to advertise them in a registry capable of fine-grained semantic matchmaking. Universal Description, Discovery and Integration (UDDI) is a Web-based distributed registry for the SOA [3]. It is one of the central elements of the interoperable framework and an OASIS standard with major backers including IBM and Microsoft. However, UDDI is limited to using flat syntax-based identification and classification system. It is not capable of storing and processing semantic service descriptions written in OWL.

It is clear that semantic annotation and matchmaking for Web services will produce much more refined search results than UDDI-style syntactic matching [4, 5]. It is also clear that UDDI is fast becoming widely accepted as a Web infrastructure standard already with widespread deployment by companies, government agencies, and the military. The goal of this ongoing work is to add OWL based semantic markups and query capabilities to existing registry implementations that conforms to the UDDI V3 specification. Service descriptions can be expressed using the OWL-S ontology, however, our approach provides support for the OWL language as a whole and any ontology can be used. This approach does not require any modification to the

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 89 – 98, 2005.
© Springer-Verlag Berlin Heidelberg 2005

existing UDDI infrastructure. Users that wish to take advantage of semantic annotation and query capabilities can simply install modules in their own client machines and use UDDI registries as semantic registries. This will not be the ideal solution in the long term. Registries specifically developed for semantic service description and query processing will be much more effective and efficient. However, this approach will provide a short-term solution that will allow organizations to start using OWL service descriptions without having to make significant additional investments in SOA infrastructure.

2 Semantic Annotation and Queries

This section describes the kinds of semantic annotations and queries we plan to support. We will use the following three example ontologies in figure 1 and the service description concept in figure 2 throughout the rest of the paper.

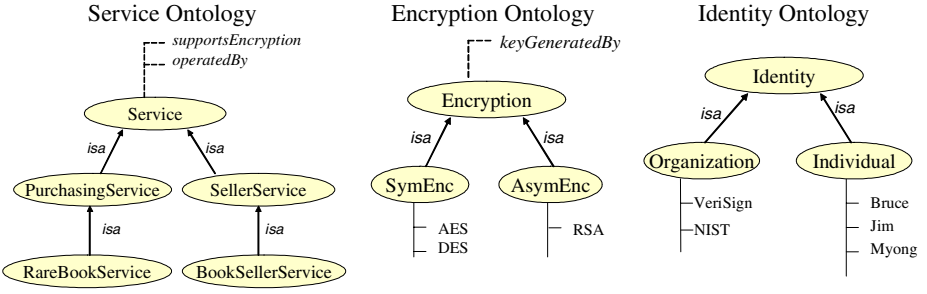


Fig. 1. Ontology examples. Ovals represent classes, solid lines represent instances, dotted lines represent properties, and arrows represent subclass relationships.

Semantic annotations describe Web services using concepts from ontologies. For example, a Web service may advertise itself as a BookSellerService from the Service Ontology. The service description can further annotate the ontology concept by defining additional properties on BookSellerService such as in figure 2.

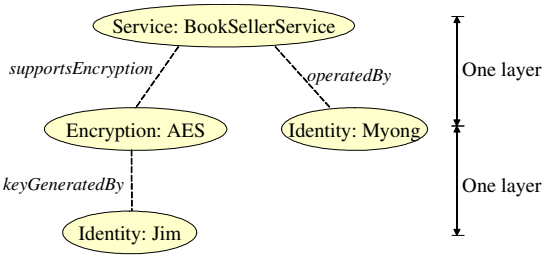


Fig. 2. An example of Web service semantic annotation

Semantic annotations can have multiple layers of annotations. However, the UDDI data model is only capable of storing one layer of annotations because it was designed to deal with flat identification and categorization systems. Thus the first challenge is to correctly express complex semantic service descriptions in the UDDI data model without losing any details.

The second challenge is to provide support for semantic query. Two types of queries must be supported by the system.

Exact Match Queries: The first type of queries uses the same concepts as those specified in the service description. Query processing does not require ontology awareness.

- Find a BookSeller service
- Find a BookSeller service operated by Myong
- Find a BookSeller service that supports AES encryption
- Find a BookSeller service that supports AES encryption with the key generated by Jim.

Semantic Match Queries: The second type of queries uses semantically related concepts as those specified in the service description. Query processing requires ontology awareness.

- Find a BookSeller service that supports SymEnc
- Find a BookSeller service that supports Encryption
- Find a BookSeller that supports SymEnc with the key generated by Jim.

The classes and instances specified in these queries are not the same as those in the service description. However, they should all match to the concept in figure 2 because they are related due to the class hierarchy and inferences established by the ontologies. This semantic information must be captured and processed by the UDDI registry in order to support semantic matchmaking.

3 Mapping Strategy

UDDI is intended to serve as a repository for Web services. The UDDI specification defines a set of core data models for storing Web service descriptions and an API for interacting with the registry. The core data model consists of objects describing the Web service (*businessService*), the service provider (*businessEntity*), and the service binding (*bindingTemplate*). In addition to the static text-based data fields, these data objects can incorporate metadata into the description by making references to *tModels*. TModels, or technical models, provide extensibility to the overall UDDI framework because they can be used to represent *any* concept or construct. The versatility of the tModel comes from the fact that it only serves as a place holder. Information is not actually imported into the registry. UDDI core data model objects reference tModels with *keyedReferences* and *keyedReferenceGroups*. KeyedReferenceGroups can be used to group logically related concepts.

The tModel framework is very powerful and provides the UDDI system with a great deal of extensibility and flexibility. They can be used for a variety of different purposes including for the storage of ontology information.

In this mapping scheme, each individual concept within the ontology including all instances, classes and properties will be incorporated into the UDDI registry as tModel objects that can be referenced individually. This use of tModels is much more complex than what is envisioned by the UDDI specification. Each distinct ontology concept will be represented by a separate tModel. Additional tModels will be created to represent anonymous composite concepts defined in service descriptions such as the one in figure 2. Figure 3 shows the UDDI representation of that concept using tModels and keyedReferences.

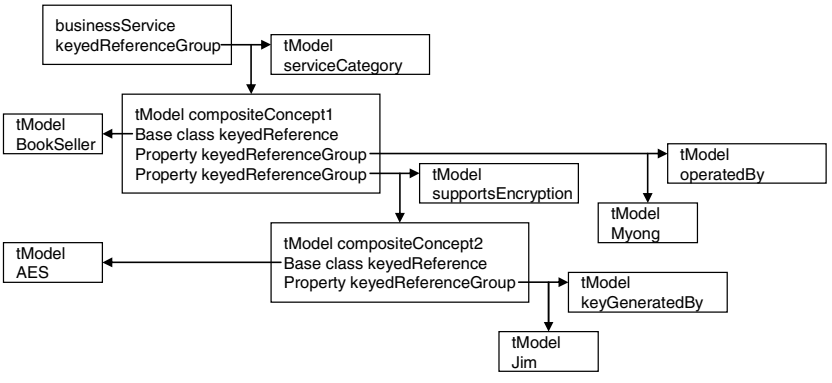


Fig. 3. UDDI tModel representation of the complex service description in figure 2

The overall composite concept is represented by the tModel "compositeConcept1." In addition, the component composite concept, AES with key generated by Jim, is represented by its own tModel "compositeConcept2." Composite concept tModels will hold a reference to designate its base class. Annotations are captured as keyedReferenceGroups what will reference the property type and property value tModels as children keyedReference elements. This way, each tModel can hold one layer of annotations and chaining multiple tModels together will allow for representation of composite concepts with multiple layers of annotation. The entire composite concept can be referred to by referencing the top layer concept tModel.

4 Semantic Support

The UDDI search engine is only capable of performing syntax matching. However, OWL requires a semantic matchmaker capable of taking into account relationships between concepts established by the ontology. Our approach fully resolves and indexes ontology relationships at publishing time of the ontology. This way, queries can be processed syntactically by the UDDI search engine and yield results equivalent to those of a semantic matchmaker.

The first type of ontology relationships that must be resolved involves the property and class hierarchy defined using the *subClassOf*, *equivalentClass*, *subPropertyOf*, and *equivalentProperty* constructs. *Identity concepts* are defined as

the set of related concepts for which queries should yield the original concept based on the class and property hierarchy established in the ontology. If query for Encryption should return class RSA, then Encryption would be an identity class of RSA. An ontology reasoner can resolve the list of identity concepts for all the base concepts at the publishing time of the ontology. When service descriptions are published, any references made to ontology concepts need to also include the identity concepts. For example, if a service is capable of RSA, it also needs to indicate that it is capable of Encryption. This way, queries for both RSA and Encryption will yield the service.

The second type of relationships that must be resolved involves property characteristics. Object properties can be defined with the characteristics of *TransitiveProperty* and *SymmetricProperty*. *Inferred properties* are properties not explicitly defined in the ontology or service description but could be inferred based on property characteristics and other property definitions. Inferred properties will be resolved for all the base ontology concepts by the reasoner at the publishing time of the ontology. For symmetric properties, the reverse of the explicit property definition must also be defined as an inferred property. For transitive properties, the entire chain of transitivity must be resolved and referenced as inferred properties. When service descriptions are published, it must make appropriate references to inferred properties in addition to the explicit property definitions.

True ontology awareness is only necessary during publication of the ontology. Publication and query of service description can be done syntactically using identity concepts and inferred properties captured during ontology publication. The UDDI search engine, however, is not capable of all the syntax matching operations necessary due to its lack of supporting Boolean queries. Therefore, our system will use an additional matchmaker component on the client side. The part of the query processed by the UDDI search engine will only deal with base ontology class of composite concepts and return a coarse list of possible matching services. The matchmaker on the client side will refine the list by matching composite concepts in their entirety. For example, if the query is for the composite concept presented in figure 2, the query passed on to UDDI will simply be for BookSeller. The matchmaker on the client side will match the service descriptions returned by UDDI against the full composite concept.

Our prototype implementation will not fully support all aspects of the OWL language [1]. This is governed by the functional limitations of UDDI as well as the desire to keep the prototype relatively simple. The system will not enforce validation of ontologies and service descriptions. It will be up to the user to validate their own OWL documents. Complex class expressions involving *intersectionOf* and *complementOf* will not be supported. This is the lack of Boolean query support in the current version of UDDI specification. Class expressions involving only *unionOf* will be supported because the identity concept approach treats sets of classes and properties as unions by default. More advanced inferences based on cardinality, complex class expressions and other property characteristics and will not be supported. These types of inferences are generally intended for reasoning about the ontology and are not directly relevant to matchmaking. It is important to note that the actual mapping is lossless and all information will be captured inside UDDI data structures. The limitations are on the query side in that some constructs will not be taken into consideration during query processing.

5 System Architecture and Query Processing

Details of our prototype implementation can be found in [6]. Figure 4 shows the overall system architecture. The shaded boxes are the four add-on modules that will reside in client sides. Only clients wishing to use UDDI registries as semantic registries will need to add these modules to their machines.

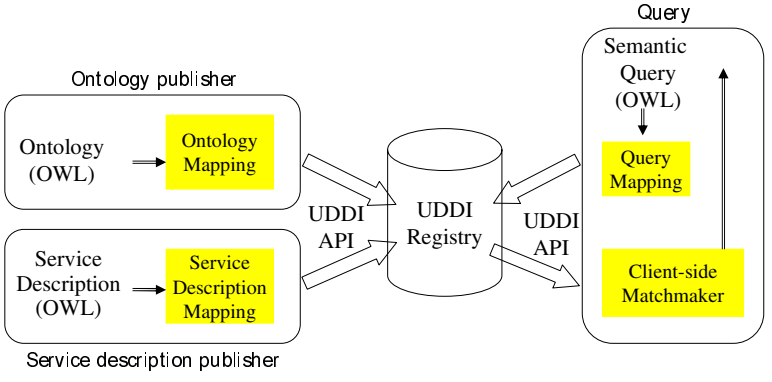


Fig. 4. A system architecture for semantic processing with UDDI

Ontology mapping and service description mapping modules are implemented in XSLT and Java. XSLT translates OWL documents into the UDDI data model and Java code publishes them into the registry using the UDDI client-side API. Identity concepts and inferred properties are resolved in the ontology mapping module which includes a simple ontology reasoner implemented in XSLT. The service description mapping module will syntactically propagate semantic information captured in the ontology tModels to the service descriptions.

The query mapping module is also implemented in XSLT and Java. The XSLT component will strip annotations from composite concepts and translate the OWL queries into the UDDI queries. The Java component will then query the registry through the UDDI client-side API. The results returned by the registry will be the businessServices objects with references that match only the base ontology concepts.

The client-side matchmaker is a Java module that will refine the query results by performing matchmaking that takes into account all the annotations of composite concepts. It will query the registry for concept tModels referenced by the businessService object to fully reconstruct the service description. Then it will match the service description with the query by examining the concepts at each layer of annotation. This component is only necessary because the UDDI specification lacks support for Boolean queries. Its task can be folded into the registry if future versions of UDDI provide that support.

6 Mapping Specification for Publication

This section summarizes the mapping specification from OWL ontology and service descriptions to the UDDI data model.

6. 1. Ontology Mapping

Ontology tModel: The ontology tModel that will serve as a place holder and namespace for the ontology as a whole. It will hold overview information including the ontology name, description, and URL of external descriptions. This tModel will be referenced by all instance, class and property tModels associated with the ontology using the KeyValue of "ontologyReference."

Property Type tModel: The property type tModel will store ObjectProperty information defined in the ontology. The name of the tModel will be set to the name of the class defined in the ontology. Information not used for query processing such as domain, range, and property characteristics will also be captured.

```
<owl:ObjectProperty rdf:ID="operatedBy">
  <rdfs:domain rdf:resource="#Service"/>
  <rdfs:range rdf:resource="&identity;Identity"/>
  <rdfs:type rdf:resource="&owl:TransitiveProperty">
</owl:ObjectProperty>
```



```
<tModel tModelKey="uuid:operatedBy">
  <name>operatedBy</name>
  <categoryBag>
    <KeyedReference keyValue="ontologyReference" tModelKey="uuid:service"/>
    <KeyedReference keyValue="true" tModelKey="uuid:IsOntologyCore"/>
    <keyedReference keyValue="domain"  tModelKey="uddi:service:Service" />
    <keyedReference keyValue="range"  tModelKey="uddi:idnetity:Identity" />
    <keyedReference keyValue="TransitiveProperty"  tModelKey="uddi:owl:type" />
  </categoryBag>
</tModel>
```

Fig. 5. tModel that maps an object property

Class and Instance tModel: The name of the tModel will be set to the name of the concept defined in the ontology. It will hold keyedReferences to the tModels representing its identity concepts. The list of identity relationships will be derived by the ontology reasoner based on the class hierarchy. The types of the identity concepts are stored in the keyValue field of keyedReferences. All classes will have an identity relationship to itself with the relationship type of "exactRelationship." The other possible relationship types are "equivalentRelationship" for equivalent concepts, "generalizationRelationship" for parent concepts, and "specializationRelationship." for children concepts. Classes defined as union of other classes will hold identity concepts to those other classes as well as their identity concepts.

```
<owl:Class rdf:ID="SellerService">
  <rdfs:subClassOf rdf:resource="#Service"/>
</owl:Class>
```



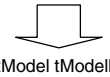
```
<tModel tModelKey="uuid:SellerService ">
  <name>SellerService</name>
  <categoryBag>
    <KeyedReference keyValue=”ontologyReference” tModelKey=”uuid:Service”/>
    <KeyedReference keyValue=”exactRelationship” tModelKey=”uuid:SellerService”/>
    <KeyedReference keyValue=”generalizationRelationship” tModelKey=”uuid:Service”/>
    <KeyedReference keyValue=”specializationRelationship” tModelKey=”uuid:BookSeller”/>
  </categoryBag>
</tModel>
```

Identity concepts

Fig. 6. tModel that maps the SellerService class from figure 1

Composite Concept tModel: Composite concepts are classes and instances with property definitions. They can be defined in the ontology as restriction classes or in service descriptions as anonymous instances. For restriction classes, the tModel name will be set to the name of the class defined in the ontology. Anonymous instances are not named and the name of the tModel will be left blank. Composite concepts can have multiple layers of annotations. If the annotations are composite concepts themselves, new tModels need to be created for them as well. Property definitions are captured as keyedReferenceGroups. The property type and property value tModels as well as the tModels of their identity concepts are captured as keyedReferences under the keyedReferenceGroup. Both explicit and inferred properties are captured the same way and no distinction is made between them.

```
<owl:Class rdf:ID="SecureSellerService">
  <rdfs:subClassOf rdf:resource="#SellerService"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="supportsEncryption"/>
      <owl:hasValue>encryption:AES</owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```



```
<tModel tModelKey="uuid:SecureSellerService ">
  <name>SecureSellerService</name>
  <categoryBag>
    <KeyedReference keyValue=”ontologyReference” tModelKey=”uuid:service”/>
    <KeyedReference keyValue=”true” tModelKey=”uuid:IsOntologyCore”/>
    <KeyedReference keyValue=”exactRelationship” tModelKey=”uuid:SellerService”/>
    <KeyedReference keyValue=”generalizationRelationship” tModelKey=”uuid:Service”/>
    <KeyedReference keyValue=”specializationRelationship” tModelKey=”uuid:BookSeller”/>
    <KeyedReferenceGroup tModelKey=”uuid:propertyDefinition”>
      <KeyedReference keyValue=”exactProperty” tModelKey= uuid:supportsEncryption”>
      <KeyedReference keyValue=”exactRelationship” tModelKey=”uuid:AES”/>
      <KeyedReference keyValue=”generalizationRelationship” tModelKey=”uuid:SymEnc”/>
      <KeyedReference keyValue=”specializationRelationship” tModelKey=”uuid:Encryption”/>
    </KeyedReferenceGroup>
  </categoryBag>
</tModel>
```

base class

Identity concepts

property type

property value

Identity concepts

Fig. 7. tModel that maps a restriction class

6.2 Service Description Mapping

Translation of service description involves two steps. First, tModels for any anonymous composite concept defined in the service description must be published into the registry unless they already exist. Second, the service description must be translated into a corresponding UDDI businessEntity and businessService objects.

Service Description: Information that maps to the UDDI businessEntity and businessService data model objects can be translated directly. Ontology references in the service description are stored as keyedReferenceGroup the same way as properties in the tModels for composite concepts. In addition, the base class of composite property value concepts is also referenced.

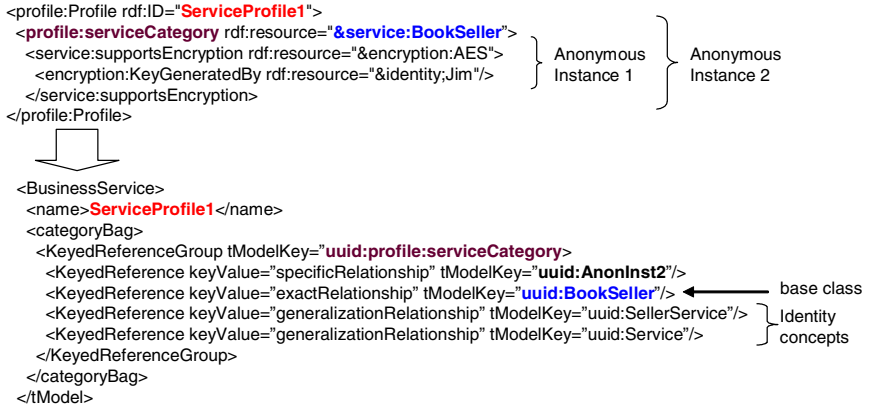


Fig. 8. BusinessService referencing the concept in figure 2

7 Related Work

Srinivasan [7] added semantic support to UDDI by placing add-on modules on the registry side. This means the existing registry infrastructure needs to be modified extensively to provide semantic support. Furthermore, the add-on modules create special interfaces for processing semantic publications and queries separate from the UDDI interface. In effect, these modules act as separate semantic registries that happen to be on the same server as opposed to integrated with the UDDI registry.

Sivashanmugam [4] developed a scheme to store ontology-based semantic markups using the native UDDI data model. However, their solutions do not support composite concepts with multiple layers of annotations. Ontology concepts can be referenced as is, but they cannot be further annotated by service descriptions. Furthermore, class hierarchy between concepts in the ontology is not captured by the translation. It is not clear if semantic queries can be supported using this approach.

The Web Service Modeling Ontology (WSMO) based Web Service Modeling Language (WSML) is an alternative to OWL-S and OWL for semantically describing

web services [8]. Since a direct mapping exists between WSML and OWL [9], WSMO will be supported indirectly in our system.

8 Conclusion

We presented an approach for supporting semantic markups of Web services and semantic queries using existing registries conforming to the UDDI V3 specification. Support is provided for the OWL language as a whole and the system will operated with any OWL ontology including OWL-S. A special lossless translation scheme that fully supports composite concepts was developed to store ontologies and semantic service descriptions inside UDDI registries. Once all the semantic information is captured, semantic query processing can be performed using a combination of the UDDI search engine and syntax based client-side matchmaker.

This approach does not require any modification to the existing registry or infrastructure. The advantage is that it is completely backward compatible. The add-on modules only need to be installed on the clients of users who wish to take advantage of semantic markups. They can be integrated seamlessly into existing systems and operations without any modification of the infrastructure.

References

1. W3C, "OWL Web Ontology Language Overview." 2004 <http://www.w3.org/TR/owl-features/ >.
2. Web Ontology Working Group, "OWL-S: Semantic Markup for Web Services," W3C. <http://www.daml.org/services/owl-s/1.1/overview/>.
3. UDDI Spec Technical Committee, "UDDI Version 3.0.2," OASIS. 2004 <http://uddi.org/pubs/uddi_v3.htm>.
4. K. V. K. Sivashanmugam, A. Sheth, and J. Miller, "Adding Semantics to Web Services Standards," presented at International Conference on Web Services, 2003.
5. A. Dogac, G. Laleci, Y. Kabak, and I. Cingil, "Expoliting Web Service Semantics: Taxonomies vs. Ontologies," IEEE Data Engineering Bulletin, vol. 25, 2002.
6. J. Luo, B. Montrose, and M. Kang, "Adding Semantic Support to Existing UDDI Infrastructure," Naval Research Lab, Washington, D.C., NRL Memorandum Report NRL/MR/5540-05-650, 2005.
7. M. P. N. Srinivasan, and K. Sycara, "Adding OWL-S to UDDI, implementation and throughput," presented at First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004), San Diego, California, USA, 2004.
8. W3C, "Web Service Modeling Ontology (WSMO)." 2005 <http://www.w3.org/Submission/WSMO/>.
9. W3C, "Web Service Modeling Language (WSML)." 2005 <http://www.w3.org/Submission/WSML/>.

A Multiagent-Based Approach for Progressive Web Map Generation*

Nafaâ Jabeur and Bernard Moulin

Computer Science Department, Laval University, Ste-Foy, G1K 7P4 Québec, Canada
Geomatic Research Center, Laval University, Ste-Foy, Québec G1K 7P4, Canada
{nafaa.jabeur, bernard.moulin}@ift.ulaval.ca

Abstract. Demands for web mapping services are increasing worldwide since the rise of Internet which became a growing medium to disseminate geospatial information. However, these services need to be more reliable, accessible and personalized. They also need to be improved in terms of data format, interoperability and on-the-fly processing and transfer. In this paper we propose a multiagent-based approach to generate maps on-the-fly in the context of web mapping applications. Our approach is abls to adapt the contents of maps in real-time to users' needs and display terminals. It also speeds up map generation and transfer. Our approach which is called *progressive automatic map generation based on layers of interest*, combines different techniques: multiagent systems, cartographic generalization and multiple representations.

1 Introduction

Thanks to new advances in communication technologies, development standards and information storing and handling techniques, more and more mapping services are disseminated over the Internet for different needs such as positioning, tourism, transport management, emergency and military use [1]. Currently, the approach used to provide such services consists in retrieving and displaying preprocessed spatial data from a database stored in a specialized server. In addition to the map quality which diminishes during scale reduction, this approach lacks facilities to personalize the map with respect to the user's needs, preferences and context. Therefore, we need to apply the appropriate cartographic generalization[1] operators (transformations) on-the-fly to spatial objects [3] in order to adapt the contents of maps in real-time to users' needs and display terminals.

On-the-fly web mapping can be defined as the real-time creation and the immediate delivery of a new map upon a user's request and according to a specific scale and purpose. In order to provide on-the-fly web mapping services, we have to deal with different kinds of constraints. Indeed, in addition to data processing, transfer and

* This work was funded by GEOIDE (Réseau de centres d'excellence en géomatique) and supported by the Ministry of natural resources of Québec, the Center of Research of Defense at Valcartier (Québec), the Center of Topographic Information at Sherbrooke (CITS).

[1] Cartographic generalization can be defined as the science and the art to exaggerate the important aspects (entities) in accordance with the purpose and the scale of a particular map and the exclusion of non-pertinent details that may overload the map and confuse its user [2].

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 99 – 108, 2005.
© Springer-Verlag Berlin Heidelberg 2005

visualization, the map generation process must be automatic and has to take into account users' reading and reasoning abilities. We also need a rigorous approach that enables us to speed up real-time map generation and transfer.

In this paper, we propose a multiagent-based approach to generate maps on-the-fly in the context of web mapping applications. Our approach aims to emphasize objects which are important to users and to adapt in real-time maps' contents to users' needs and preferences. In Section 2, we identify the different kinds of constraints of web mapping services. In section 3, we present how we structured and categorized our data in order to generate the required maps. In Section 4, we present the multiagent approach that we propose to generate maps on-the-fly in the context of web mapping applications. In this section, we present the multi-layer architecture of our multiagent system, how spatial objects are processed and how maps are generated. In Section 5, we present our innovative approach to generate maps on-the-fly: it is called *progressive automatic map generation by layers of interest*. A layer of interest refers to a collection of spatial objects which have the same importance to the user. This collection may encompass data from different classes of objects such as roads, buildings and lakes. Finally, in Section 6, we present an application in the tourist domain.

2 Constraints of Web Mapping Services

On-the-fly web mapping is a challenging task. It has to deal with four kinds of constraints: *technical constraints*, *spatial data constraints*, *user constraints* and *spatial processing constraints*. Technical constraints are independent of the approach used to generate the required map. They result from downloading time, data transfer rates, screen sizes and resolution, etc. [4]. Spatial data constraints are related to data modeling, availability and retrieval. Furthermore, users' constraints result from users' requirements, preferences, cultural backgrounds, contexts and spatial reading and reasoning abilities. Spatial processing constraints are related to automatic cartographic generalization and real-time map generation. Indeed, when trying to generate a map, it is still a challenging task to choose relevant generalization operators, their particular implementation algorithms and the best sequence in which to apply them. Furthermore, spatial processing has to make an efficient use of spatial data. It needs to adapt the content of maps in real-time in order to support users and technical constraints. During this process, spatial conflicts may occur and have to be solved.

Since generalization operators are often time-consuming and therefore may not be suitable to on-the-fly map generation, the idea is to use multiple representation to tackle this time-consuming problem and to provide better personalized map contents to users. Multiple representation can be defined as representing the same geographic entity in different ways in the same system [5]. Indeed, depending on the purpose of the map and on its intended usage, representing certain objects by their graphic and/or semantic representations may be more significant to users than using their geometric representations. In this case, users can better interpret the map and reason about its content. The challenges related to the use of multiple representation concern the choice of the best representations of objects, the enrichment of the database by the different representations and its maintenance. Due to the large number of constraints,

it is important to prioritize the issues to be tackled for web map generation. In this paper, we are particularly interested in finding ways to provide maps whose contents can be adapted on-the-fly to users' needs and to their display terminals. We are also interested in finding ways to speed up the real-time generation and transfer of maps.

3 Spatial Data Model and Structure

In order to provide a map content adapted to users' needs, preferences and context, it is important to set up an appropriate spatial ontology. In a related work [12], we proposed an ontology for the spatial object category *Building* in a tourist context. It includes categories such as *Accomodation*, *Restaurant*, *Leisure*, *Transport*, *Emergency* and their sub-categories. This ontology can be easily enriched to encompass new categories related to the road network, the hydrographic network, plazas, etc. The spatial objects are stored in GML files according to this ontology are.

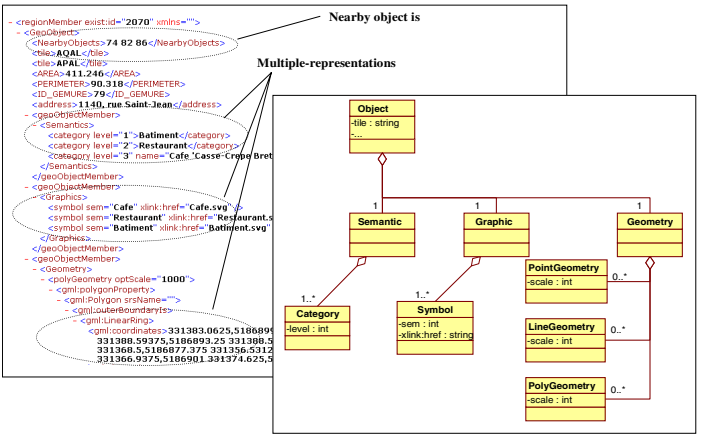


Fig. 1. The structure of multiple representation in a GML file and in a UML diagram

In order to structure the data associated to spatial objects, we assign three representation categories to each spatial object: a geometric representation, a graphic representation and a semantic representation. Each representation category may have several instances (Figure 1.b). By combining any triplet instances of the three representation categories of a given object, we get a new representation of this object. Furthermore, in order to speed up the web map generation, we pre-calculate and store the nearby objects of every spatial object. This information is important when overlaps appear between spatial objects.

In order to improve the personalization and the usability of web maps, the system focus on the information that most interests users. Our idea is to categorize spatial objects into different level of importance also called *layer of importance* or *layer of interest*. A layer of interest refers to a set of spatial objects that interest the user and which have the same importance to him. These objects may belong to different

classes of objects such as roads, specific buildings, lakes and places. Data categorization enables us to define different priorities (weights) that we assign to spatial objects according to their importance for the user. This categorization may vary according to the purposes of the map to be generated and to its context of use. For example, in a tourist context, we can categorize spatial objects into *explicitly required objects*, *road network*, *landmarks* and *ordinary objects* (objects which do not have any specific importance for the tourist). However, in a military context, in addition to the road network, the hydrographic network and buildings layers, we may need a layer of interest that contains strategic features (buildings, places, bridges, etc.) to be protected, a layer of interest that contains friend's forces and another one that contains enemy positions.

4 A Multiagent-Based Approach for On-the-Fly Map Generation

Several research works used multiagent systems for automatic cartographic generalization [6, 7, 8, 9]. But few works combine multiagent systems cartographic generalization and multiple representations in order to generate maps on-the-fly. An exception is the SIGERT project [12] in which our research takes place. The advantages of using multiagent systems are multiple: their capability to often reach acceptable solutions, their flexibility to solve complex problems, their dynamic adaptation to environment changes and their ability to successfully model the whole process of automatic map generation. In this paper we present a multiagent-based approach for on-the-fly map generation. Our innovative approach combines three techniques: multiagent systems, cartographic generalization and multiple representation. It consists in creating and assigning software agents to spatial objects and then to let these objects interact in order to generate required maps according to users' needs and display terminals characteristics. Our multiagent system is composed of two main modules: a control module and a spatial data processing module (Figure 2).

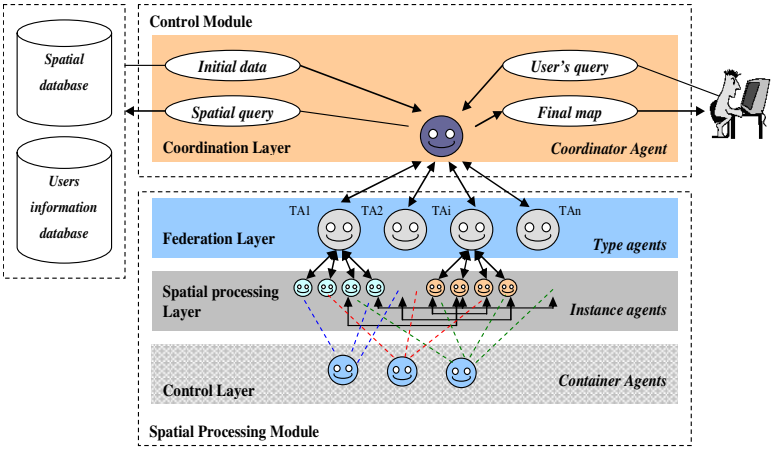


Fig. 2. Multi-layer architecture of our Multiagent system

4.1 The Control Module and the Spatial Processing Module

The ***control module*** contains a *coordinator agent* which is responsible for the communications with client applications. If a user wants to get a map, it selects from its user interface the classes of objects or the particular objects he is looking for, then his query is processed by the control module. The *coordinator agent* analyzes the user's query and extracts relevant data from the *spatial database* and the *user database*. The *users database* is necessary to authenticate users and to record parameters used to personalize maps' contents. It contains information such as the users' preferred display language and his personal perception threshold (this threshold can be determined by offering to users to select a perception threshold from a predefined thresholds).

The *coordinator agent* splits the retrieved data into several datasets according to the spatial data categorization (each dataset concerns a given layer of interest), and then sends these datasets to the *spatial processing module*. As soon as the *coordinator agent* receives the final data of a given layer of interest from the spatial processing module, it carries out a final adaptation of this data in order to improve its personalization and transfers it to the user's terminal for display.

The ***spatial processing module*** is composed of three layers. The first layer is the *federation layer* which contains several *Type agents*. Each *Type agent* is assigned to a specific layer of interest. It creates and assigns an *instance agent* to each spatial object of its type. The second layer of the *spatial processing module* is called the *spatial processing layer*. It contains all the *instance agents* created by the different *Type agents*. The *instance agents* are responsible for the generation of the contents of the required maps. The third layer of the *spatial processing module* is the *control layer*. It is composed of *container agents*. A *container agent* encompasses all the objects that should be aggregated in the map at a scale inferior to the scale of the map required by the user. The importance of container agents lies in the acceleration they give to the map production process. Indeed, when *instance agents* are not able to solve their conflicts due to lack of space, the *container agents* intervene and impose an arbitration solution in order to solve the deadlock conflicting situation.

4.2 Our Map Generation Approach

The *instance agents* negotiate with each other when conflicts appear in order to produce the required map. A negotiation depends on the agents' priorities as well on their environments (priorities of the nearby agents, actions performed by neighbor agents, etc.). When a conflict is detected between two agents (Figure 3), the agent with lower priority tries first to solve the conflict. If it fails to solve the conflict, then it asks the agent with higher priority to solve it. If the conflict remains unsolved, then the first agent must solve it or eliminate itself. During the agents' negotiation, objects may be displaced, scaled-down, exaggerated, merged or eliminated. Furthermore, the representation of certain objects may be changed in order to emphasize their semantics or to create free space wherein to place their symbols. At each step of map generation, every *instance agent* supervises the changes in its immediate environment and evaluates its current state (Figure 4).

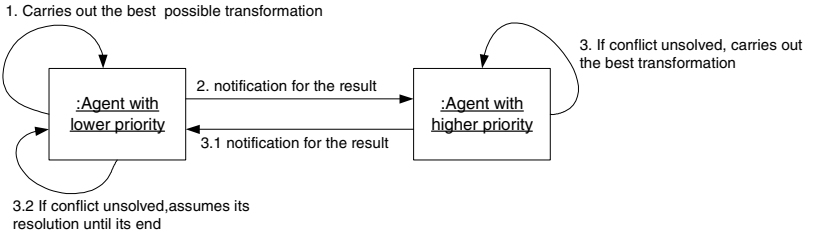


Fig. 3. Negotiation pattern between agents

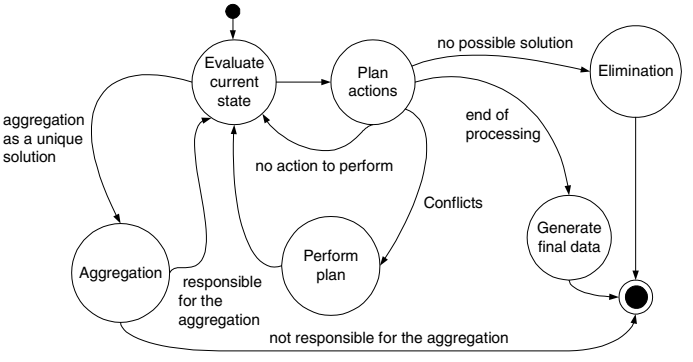


Fig. 4. Spatial agent life cycle automaton

Then, it plans and performs relevant actions. In order to choose these actions, the agent verifies the results that it may obtain by size reduction, displacement, aggregation or representation change, then chooses the best action to perform. If no solution can be found to solve a conflicting situation, the instance agent eliminates itself if it has no real importance for the user. During their interactions, agents exchange messages whose contents may be the agents' coordinates, notifications of existing conflicts or of actions carried out as well as their results, etc. Each time an agent reaches the end of its processing, it notifies its nearby agents. An agent finishes its processing when its neighbors with higher priorities finish theirs. This strategy enables us to generate the required map, layer by layer according to the importance of spatial data.

5 Progressive Automatic Map Generation by Layers of Interest

There are four fundamental approaches to provide on-the-fly web maps. The first approach is *representation-oriented* (Figure 5a). It is based on a multi-scale database which is computed off-line. This approach allows real-time map generation but lacks flexibility with respect to users' needs and preferences. The second approach is *process-oriented* (Figure 5b). It relies on real-time map generation. This approach is very flexible. However, it is not widely used because of the time it takes to create the

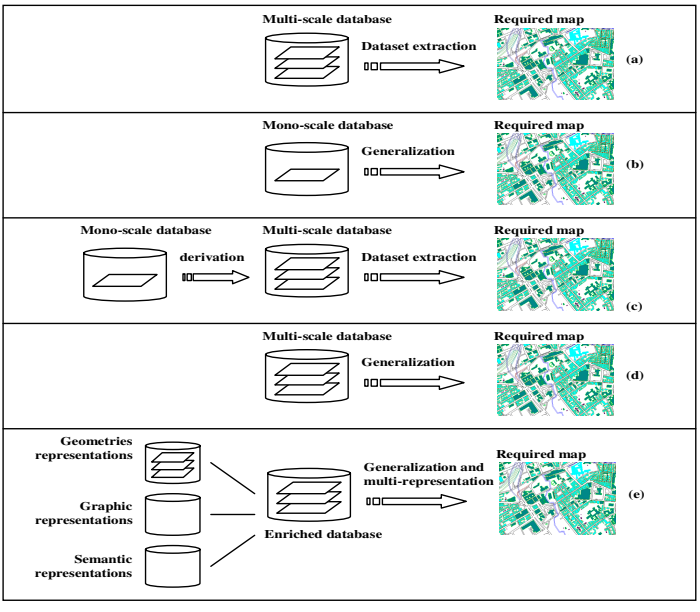


Fig. 5. Different approaches for on-the-fly map generation

required maps. The third approach is *derivation-oriented* (Figure 5c). It is based on a multi-scale database whose levels are derived from one detailed database. This derivation lacks flexibility and requires a lot of time and human resources. The fourth approach combines the use of multi-scale database and generalization. This approach (Figure 5d) takes advantage of the flexibility of a process-oriented approach and the suitability of a representation-oriented approach to real-time map generation. However, it lacks flexibility since it uses predefined sequences of operators in order to generalize different data types.

The approach that we propose (Figure 5e) is called *progressive map generation by layers of interest* and consists in the enrichment of the spatial base with semantic and graphic representations in addition to geometric representations, then in the generation of the maps using cartographic generalization and multiple representation. The generation of these maps is done by our multiagent system that first produces and then transfers the final map, layer by layer according to the data categorization (Figure 6). The first transferred layer contains the most objects which are most important to the user. The use of a progressive data transmission approach in the context of

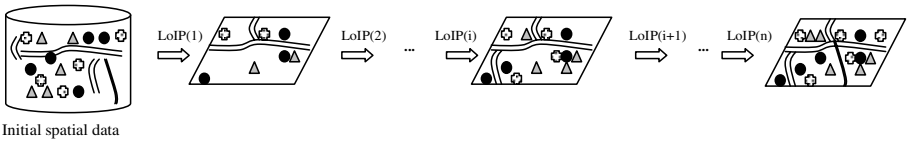


Fig. 6. Progressive automatic map generation by layers of interest (LoIP(i): Layer of Interest of Priority i)

web mapping services already exists. The formerly proposed approaches [10, 11] tackle low rates and high delays of data transfer. However, users are still obliged to wait until whole maps are downloaded. In our approach, as soon as all the objects of a given level of importance are generated, the correspondent layer of interest is generated and then transferred to the user. Our approach also manages low rates and high delays of data transfer and shortens users' waiting time. Indeed, the user can stop the transfer of data without waiting the whole download of the map, as soon as he gets the required information from the data already transferred.

6 Application: The SIGERT System

We tested our multiagent system approach in the context of the SIGERT project (Figure 7) which aims to provide maps for web and mobile users on the basis of multiple-representations and cartographic generalization [12].

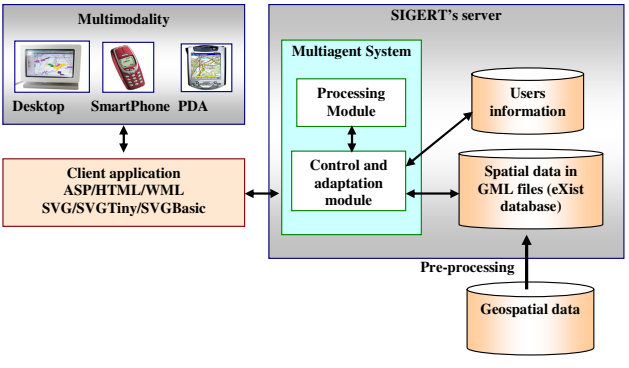


Fig. 7. Architecture of SIGERT

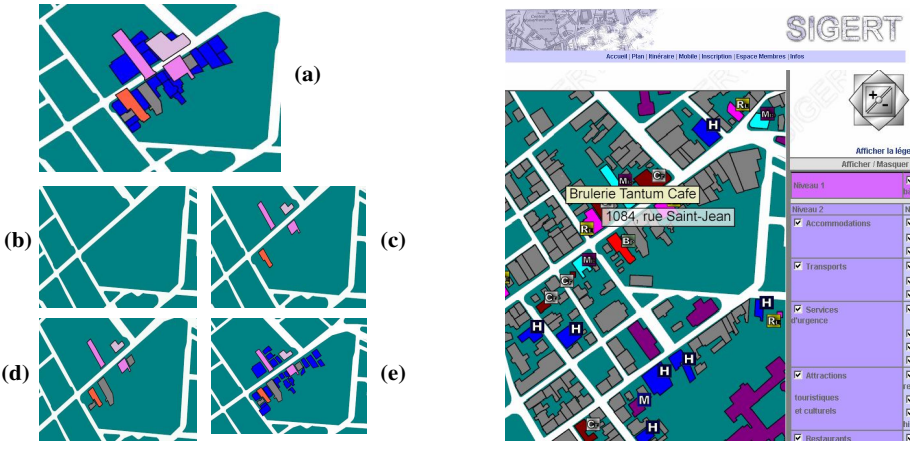


Fig. 8. [left] (a) Initial configuration with spatial conflicts, (b) RN layer, (d) RN and ERO layers, (e) RN, ERO and LMO layers, (f) Final configuration [right] example of web maps

We developed our multiagent system using the *Jade* platform [13]. Our data have been categorized into four categories: *explicitly required objects (ERO)*, *landmark objects (LMO)*, *road network (RN)* and *ordinary objects (OO)*. Spatial data are stored in GML files. At the end of processing, GML files are transformed into SVG files which can be displayed on users' terminals. Example of the results returned by our system are displayed in Figure 8. If the user moves the mouse over an important object in the generated web map (Figure 8 right), the textual description of this object is displayed. The user can get the address of this object by clicking on it.

7 Conclusion

In this paper we proposed a multiagent-based approach to generate maps on-the-fly in the context of web mapping applications. Our approach enables us to improve maps personalization by adapting in real-time their contents to users' needs and display terminals. The handling of spatial objects is done in an innovative way by using different techniques: multiagent systems, multiple representations and cartographic generalization. The multiagent system generates and transfers data to the user according to its importance. It uses our new approach of progressive map generation by layers of interest. This approach speeds up the generation and the transfer of data, which represents an important advantage especially in a mobile context. Our approach was tested in a tourist domain but can be extended and used in other domains in order to generate maps for other needs such as military applications and emergency management.

Currently, our prototype is still slow with respect to acceptable delays of real-time map generation. This is due to several factors: the slowness the Java language and the Jade platform; the required time to parse GML files and of the need to optimize our code. Furthermore, our prototype is also slow compared to other existing commercial web mapping system since we carry out a real-time map generalization process which is not supported by any other existing system. Future works will focus on the improvement of the system performance in terms of processing time and data visualization.

References

1. Jabeur, N., Moulin, B.: A multiagent based approach for the resolution of spatial conflicts. European Workshop on Multi-Agent Systems (EUMAS), 333-343, Barcelona (2004),
2. Hardy P. G.: Map Production From An Active Object Database, Using Dynamic Representation and Automated Generalisation. *The Cartographic Journal*, Vol. 35 No. 2, 181-189, December 1998, UK (1998)
3. Jabeur, N., Moulin B., Gbei, E.: Une approche par compétition d'agents pour la résolution de l'encombrement spatial lors de la généralisation automatique des cartes. Journées Francophones des Systèmes Multiagents JFSMA-2003, 161-173, Tunis (2003)
4. Arleth, M.:. Problems in Screen Map Design. In : Proceedings of the 19th ICA/ACI Conference, 849-857, Ottawa (1999)
5. Devogele T., Timpf S.: New tools for multiple representations. ICC'97, Stockholm, editor: Lars Ottoson, 1381-1386 (1997)

6. Ruas, A.: Modèle de généralisation de données géographiques à base de contraintes et d'autonomie. Thèse de doctorat. Université de Marne La Vallée (1999)
7. Duchêne, C., Cambier, C.: Cartographic Generalization Using Cooperative Agents. In: Proceedings of AAMAS. ACM Press, 976-977, Melbourne (2003)
8. Lamy, S., Ruas, A., Demazeau, Y., Jackson, M., Mackaness, W., Weibel, R.: The application of Agents in Automated Map Generalisation. In proceedings of 19th ICA meeting, 160-169, Ottawa (1999)
9. Maozhen, L., Sheng, Z., Jones, Ch.: Multi-agent Systems for Web-Based Map Information Retrieval. In: M. J. Egenhofer and D.M. Mark (Eds.): *GIScience*, Springer-Verlag, 161-180, Berlin (2002)
10. Buttenfield, B.: Transmission Vector Geospatial Data Across the Internet. In: M.J. Egenhofer and D.M Mark (editors), Second International Conference, GIScience 2002, Lectures Notes in Computer Science, 51-64. Springer, Berlin (2002)
11. Bertolotto, M. Egenhofer, M. J.: Progressive Transmisson of Vector Map Data over the World Wide Web. *GeoInformatica*, 5 (4), 345-373 (2001)
12. Gbei, E., Moulin B., Cosma I., Jabeur N., Delval, N. : Conception d'un prototype de service Web géolocalisé appliqué à l'industrie récréo-touristique. *Revue internationale de géomatique* . Vol. 13, 375-395 (2003).
13. Jade, JADE Project Home Page. Available at http://sharon.cselt.it/projects/jade (2005)

Semantics of Agent-Based Service Delegation and Alignment

H. Balsters, G.B. Huitema, and N.B. Szirbik

University of Groningen, Faculty of Management and Organization,
P.O. Box 800 9700 AV Groningen, The Netherlands
{h.balsters, g.b.huitema, n.b.szirbik}@rug.nl

Abstract. In this paper we concentrate on conceptual modeling and semantics of service delegation and alignment in information systems. In delegation, a source company wishes to hand over parts of its functionality together with related responsibilities to a supplying party. From the side of the outsourcer the search for a suitable supplier mostly will be a manual process with all the consequences of a long time to market, as well as trial and error before a good fit is obtained between both related parties. This paper addresses an agent-based solution for improving this match-making process in B2B markets. Part of the match-making process will be the alignment of business processes on the side of the outsourcer as well on the side of the supplier. We will provide a formal means to ensure that the delegation relationship, determined by a ruling service level agreement (SLA), satisfies specific correctness criteria. These correctness criteria are defined in terms of consistency and completeness between the delegated operation and the associated operation offered by the supplier. Our correctness criterion will concern mappings between an existing delegator schema and an existing supplier schema, and will address both semantical and ontological aspects pertaining to delegation and alignment. Agent-based delegation together with formal specifications can prove their value in the process of constructing delegation contracts. Our analysis will be performed within the modeling framework based on the UML/OCL formalism. The concepts we discussed in this paper are illustrated by an example of companies delegating billing services to Billing Service Providers.

1 Introduction

Today many companies consider delegation of services or functionality outsourcing as a key element in their business strategy [13]. By delegating. many organizations will have a better focus on their core operations and can deal with variable operational costs in stead of fixed costs. The choice of a capable outside supplier is critical, however, since such outside suppliers result in serious formal commitments based on contractual agreement. Delegation of services can be divided in two main phases: the pre-contracting phase up to the moment a preferred supplier is found (i.e., till the contract is drawn up), and the execution phase thereafter. Once selecting an outside supplier has taken place, it is often still not clear whether an appropriate match in functionalities has indeed been achieved. This paper proposes a position by envisaging a future situation for companies wishing to delegate certain functionalities,

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 109 – 120, 2005.
© Springer-Verlag Berlin Heidelberg 2005

where the process of delegation will be supported in one or more of its phases (in our case, *pre-contracting*) by an agent-based infrastructure[14,16]. We will show in this paper how an agent infrastructure enhances the process of pre-contracting, especially semantic alignment by agent-based negotiation. Our proposed framework can be seen as an extension of the classical recommender agents [11], used by customers on the Web selecting products and services from e-commerce sites. The extension here is the capability of the agents to "negotiate" in order to align semantic descriptions of the data of the outsourced function and to increase the chance that a good match between the supplier of the outsourced function will be found. Our proposed agents will keep some of the "classic" recommender agents features (like selection and ranking), but their main ability will be able to take decisions to change the constraints [7], in a confined setting mainly governed by the delegating party. Also, when matching proves difficult and alignment is apparently impossible, the agents will be able to detect this situation and could ask the human owners of the delegator and supplier agent to intervene and take external decisions. We note, however, that the scope of this paper does not include these typical recommender agent activities; we assume that these activities have already been concluded before entering the phase of alignment

In delegation, one typically has the situation that the delegated service satisfies certain input- and output requirements. These requirements will be defined in terms of the ruling service level agreements (SLAs). We will provide a formal means to ensure that the delegation relationship between delegating party and supplier, determined by a SLA, satisfies specific correctness criteria. These correctness criteria are defined in terms of consistency and completeness between the delegated operation and the associated operation offered by the supplier. Our correctness criterion will concern mappings between an existing delegator schema and an existing supplier schema, and will address both semantical and ontological aspects. Formal specifications as offered in this paper can prove their value in the setup and evaluation of delegation contracts. We will perform our analysis within the modeling framework based on the UML/OCL formalism ([10,15]). The Object Constraint Language OCL offers a textual means to enhance UML diagrams, offering formal precision in combination with high expressiveness. In [1] it has been demonstrated that OCL has at least the same expressive power as the relational algebra, (the theoretical core of the relational query language SQL), thus making OCL a very powerful language for specification of constraints, queries and views on data. Also, UML is the de facto standard language for analysis and design in object-oriented frameworks, and is being employed more and more for analysis and design of information systems ([6]). Specifying a typical SLA places high demands on the expressiveness and precision of the modeling language employed; it is in this case that OCL proves to be very effective.

This paper contributes in three ways to the theory of agent-based delegation. Firstly, we will show that abstract versions of agents (i.e., for purposes of semantic alignment), both on the side of the delegating and the supplying party, can be modeled in terms of UML/OCL. We will employ our notion of exact view to model agents; exact views can capture both the functionality aspect (calculation) and the responsibility aspect (satisfaction of input/output-constraints) of the delegated service. A SLA will be given precise specifications in terms of pre- and post-condition statements on operations in OCL, and a correctness criterion will be defined in terms

of consistency with respect to these pre- and post conditions. Secondly, we will show how to construct a mapping from delegator agents to supplier agents preserving the SLA. Such a mapping (called an *ω-mapping*) will be shown to abide to a so-called *abstract alignment schema* (called an) in UML/OCL, ensuring correct delegation of a source operation. Finally, we will show that in the framework of an ω-schema, negotiation between two agents can be described in terms of strengthening and weakening pre- and post conditions. We remark that semantic alignment of service delegation can greatly improve current trial-and-error testing methods used by delegating parties to get some guarantee beforehand that the outsourcing will be performed correctly by some preferred supplier. Testing often offers a quick and simple means to predict that the bulk of the outsourced functionality is covered correctly. In cases involving highly complex services, and high standards exist regarding quality and robustness of the outsourced services, testing (due to its non-exhaustive nature) will usually fall short in offering actual guarantees. Formal means, in combination with agent-based pre-contracting –as proposed in this paper- will then save time and prove to be more effective ([5]).

This paper is organized as follows. Section 1 offers a description of the correctness problem pertaining to alignment. In Section 2 we offer an introduction to views in UML/OCL and how they can be used to model agents.. Section 3 describes so-called *exact views* to model delegation, and alignment based on ω-schemas and ω-maps. This section also contains an illustrative example of a delegation relation. Finally, Section 4 offers conclusions and directions for further research.

2 The Problem: Ensuring Correctness of Delegation and Alignment

In delegation, one typically has the situation that a source company wishes to hand over parts of its functionality (including associated responsibilities) to an outside party. This outside party is called the supplier to which the functionality (or service) is outsourced. In terms of agents, this situation translates to a *source agent* having one or more services that will be delegated to an outside *target agent*. To be able to perform this delegation activity, one not only has to locate within the source company which operation O is to be delegated, but also all relevant attributes, relations, constraints and auxiliary operations that are used in the definition of that particular operation O. All of this source material (operation, attributes, constraints, auxiliary operations) will provide the ingredients for the construction of the source agent. On the other hand, the target company wishes to supply material to meet the requirements of the delegated operation as described in the source agent. The target agent will offer specifications of the material that the supplying company can provide.

As an example, consider the situation of a Communications Service Provider (CSP) delivering communication services, such as voice and data services to customers. This CSP wants to concentrate on its core business and therefore wishes to delegate, as a customer, its billing to a dedicated party, a Billing Service Provider (BSP). Here, billing is the business process dealing with sending invoices to end users, and requesting for payment of debts. The BSP will make up the invoices on behalf of the CSP, send the invoices to the customers of the CSP and finally collect

the money due. The collected revenues will be passed back to the CSP minus commission for the handling and taking the credit risks. For the customers of the CSP, delegation of the billing to the BSP is transparent; they do not notice that it is handled by another party and perceive no difference. Billing of a communications service is done in several steps. First the usage of the service by customers has to be accounted for in so called Usage Records (UR). With each communication session Usage Records are generated containing fields, like Origination, Destination, Connection Date: Date, Connection Start/Stop Time, Service type, etc. In our example, delegation pertains to calculating the end rating of the Usage Records, where customer specific information is used. To calculate this rating the Usage Records have to be transported to the BSP, where the records will be extended with an extra field, called *End Rate*, with domain type *Amount*. (Amount is some abstract data type consisting of two fields (number, Currency), e.g., an amount like € 5 will recorded as a pair (5, €). After calculation of End Rating, Usage Records will be transformed into Charge Records (CR). (Note that, for reasons of simplicity, we will confine ourselves to delegation of End Rating, and assume that delegation of, say, invoicing and collecting actual money can be described in a similar way.)

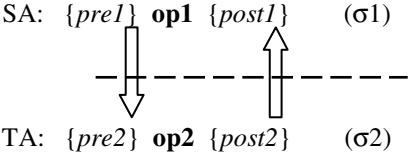
In order to perform delegation of billing functionality, the operation End Rating (denoted by `calcEndRate`) requires specific information of each service session regarding service type, the origination and destination of the communication session, the start- and ending time of the session, the volume in bytes transported in case of a data service, and possibly more. Once it has been decided that an operation like `calcEndRate` is to be delegated, one tries to locate an outside party supplying the functionality of this operation. Once such an outside (target) party is found, the source party and the target party enter negotiations regarding the quality of the outsourcing service that the target has to provide. Once an agreement has been reached, a so-called Service Level Agreement (SLA) is drawn up to which both parties are bound. A SLA is crucial in degating, since it is the sole basis on which source and target parties provide input- (responsibility of the source company) and output material (responsibility of the target company). The source agent offers the outsourced operation, say *O*, as well as all relevant attributes, relations, constraints and auxiliary operations that are used in the definition of that particular operation *O*. Furthermore, the source agent offers its initial conditions that have to be met by a target agent (offering specifications of the material that the supplying company can provide). These conditions are the basis for a SLA pertaining to source and target agents, and could typically be given in terms of pre- and post conditions. In the case of our example, we could stipulate the following conditions (written in UML/OCL).

```
context  Usage Record::calcEndRate: Amount
pre: Label = subscriber
post: Destination not in Europe implies result > 2.50 € and
StartTime > 18:00 hrs implies result < 1.00 €
```

We have assumed that Usage Records which do not relate to own subscribers of the CSP (i.e., usage records related to other roaming users) will be passed by some interface to another billing domain, and hence will be not considered here for outsourcing (pre-condition). Furthermore, this specification states that within some class

`Usage Record`, an operation called `calcEndRate` satisfies the post condition that all usage records which relate to communication sessions with a destination outside Europe are rated at least at €2.50 , and that sessions started off-peak (i.e., later then 18:00 hrs) will have a rate less than €2.50. Should one wish to delegate this operation, then the supplier is bound to this specification in terms of pre/post-conditions. This entails that the supplier is to offer an implementation `calcEndRate'` of `calcEndRate`, such that `calcEndRate'` has a pre/post- condition consistent with respect to the pre/post- condition of `calcEndRate`. For pre-conditions this means that `calcEndRate'` should not accept arguments that are not accepted by `calcEndRate`, and for post conditions it holds that `calcEndRate'` should never produce results contradicting the post condition of `calcEndRate`. Typically, a source agent is equipped with a pre/post-condition, and is roaming for a target agent that also is equipped with a pre/post-condition and that is compliant with the pre/post-condition of the supplier. This process of getting target and source agents to match in order to fulfill such compliancy, is called *alignment*.

It is the topic of this paper to offer a semantics of successful alignment. In our approach, the SLA between source and target agents provides the input for a contract binding both parties. The SLA is then used to produce a formal specification, in terms of pre- and post conditions, in which it is precisely (unambiguously) and completely stated what the supplier is expected to deliver. Such a formal counterpart of the SLA is coined a *σ-constraint*. Alignment can be described in terms of a schema. In general, in the context of delegation and alignment, source and target agents have to abide to the following (abstract) alignment schema.



(1)

This schema (called an *ω-schema*, "ω" from *o*utsourcing) is to be read as follows. SA denotes the source agent, TA denotes the target agent, σ-constraint σ1 denotes the pre/post-condition combination pertaining to the delegated operation **op1**, whereas σ-constraint σ2 denotes the pre/post-condition combination pertaining to the supplying operation **op2.** Operation **op2**, on the target side, is (by definition) a *correct implementation* of **op1**, if and only if pre-condition *pre1* logically implies pre-condition *pre2*, and post-condition *post2* logically implies post-condition *post1*. We also say that σ-constraint σ2 *is in alignment with* σ-constraint σ1. An ω-schema prescribes a consistency and completeness condition with respect to pre- and post conditions of the delegated and the supplier operation involved. In the context of an ω-schema, we can now also describe what it means that agent SA *negotiates* with agent TA in reaching agreement on a binding SLA. Initially it could be the case an SA-constraint does not align with some related TA-constraint; by negotiating, however, these two constraints could align. Typically, negotiation could be dedicated

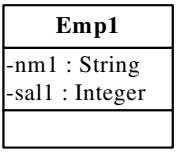
to appropriate strengthening of condition *pre1* to, say, a condition *pre1'*, and/or appropriate weakening of condition *post1* to, say, *post1'* such that pre-condition *pre1'* logically implies pre-condition *pre2*, and post-condition *post2* logically implies post-condition *post1'*. Negotiation in this sense could be the subject of semi-automatic support [13]. In this paper we focus on the problem of how to represent source and target agents such that their matching (negotiation resulting in an alignment) can be described in terms of an ω-schema. We will do so by assuming that the source and target companies can appropriately be described in terms of a UML/OCL schema, meaning that we assume that within the source company, we can provide a suitable UML/OCL model description of some operation *O* (to be delegated), as well as all relevant data, attributes, relations, constraints and auxiliary operations that are used in the definition of that particular operation *O*. This source material (operation, attributes, constraints, auxiliary operations) will provide input for the construction of the source agent, which we will describe in terms of a derived class (or *view*) with respect to the source model. On the other hand, the target company supplies similar material as input for a target agent to meet the requirements of the delegated operation as described in the source agent. Analogously, a target agent is described as a view with respect to the original model description of the target company.

Matching of source and target agents usually involves an additional step. Before we can investigate just how the matching of source and target can take place, both source and target have to be placed in the same language frame, by which we mean that we have to find a suitable mapping from model elements on the source side to model elements on the target side. Construction of such mappings from a source model to a target model belongs to the domain of so-called *data extraction*. Data extraction [4,8] deals with inconsistencies pertaining to the *ontologies* [2,3,4] of the different component information systems. Ontology studies the relation between syntax and semantics, and how to classify and resolve difficulties between syntactical representations on the one hand, and semantics providing interpretations on the other hand. Should we also wish to maintain *constraint properties* in the transition from source to target, then we move into the realm of so-called *data reconciliation*. Data reconciliation is often hard to realize, because of the severe restrictions placed on the mapping from source models to the target model. In [3,12], it has been shown that only when such a mapping satisfies certain isomorphism properties, the mapping will ensure correct resolution of the data reconciliation problem. Mapping an existing source model to an existing target model is known as the problem of *data exchange* ([9]). In general, there are no algorithms for constructing mappings solving the data exchange problem. What we can do, however, as will be done in this paper, is provide criteria by which it can be judged, *in retrospect*, whether the construction of an aligning mapping from an existing model to another existing model has been performed correctly. In our case, we will offer a criterion, formulated in terms of an ω-schema, by which we can judge that an ω-map ensures correctness of the aligning relation between source and target agents.

The next section is devoted to derived classes in UML/OCL; derived classes will be used in subsequent sections of this paper for modeling agents, and for constructing ω-maps in the context of alignment.

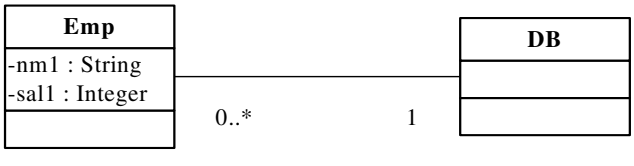
3 Views in UML/OCL and Their Application to Agent Modeling

Consider the case that in the context of some company, we have a class called Emp1 with attributes nm1 and sal1, indicating the name and salary (in euros) of an employee object belonging to class Emp1.



(2)

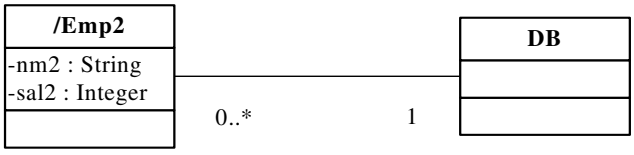
Now consider the case where we want to add a class, say Emp2, which is defined as a class whose objects are completely derivable from objects coming from class Emp1, but with the salaries expressed in cents. Assume that the attributes of Emp2 are nm2 and sal2 respectively (indicating name and salary attributes for Emp2 objects), and assume that for each object e1:Emp1 we can obtain an object e2:Emp2 by stipulating that e2.nm2=e1.nm1 and e2.sal2=(100 * e1.sal1). By definition the total set of instances of Emp2 is the set obtained from the total set of instances from Emp1 by applying the calculation rules as described above. Hence, class Emp2 is a *view* of class Emp1, in accordance with the concept of a view as known from the relational database literature. In UML terminology [20], we can say that Emp2 is a *derived class*, since it is completely derivable from other already existing class elements in the model description containing model type Emp1. Class Emp2 can be described as a derived class in UML/OCL [13,20] in such a way that it satisfies the requirements of a (relational) view. The set of instances of class Emp2 is the result of a calculation applied to the set of instances of class Emp1. The basic idea is that we introduce a class called DB that has an association to class Emp1, and that we define within the context of the database DB an attribute called Emp2. A database object will reflect the actual state of the database, and the system class DB will only consist out of one object in any of its states. Hence the variable *self* in the context of the class DB will always denote the actual state of the database that we are considering. In the context of this database class we can then define the calculation obtaining the set of instances of Emp2 by taking the set of instances of Emp1 as input.



(3)

```
context  DB
def: Emp2: Set(Tupletype{nm2:String, sal2: Integer}) =
           (self.emp1-> collect(e:Emp1 |
            Tuple{nm2=e.nm1, sal2=(100*e.sal1)}))-> asSet
```

In this way, we specify Emp2 as the result of a calculation performed on base class Emp1. Graphically, Emp2 could be represented as follows where the slash-prefix of Emp2 indicates that Emp2 is a derived attribute:



(4)

Modeling Agents as Exact Views

Exact views belong to the domain of data extraction, and are constructed from a certain collection of injective conversion functions ([2,3]). Exact views have the property that they are correctly updatable, in the sense that any update on an exact view corresponds to exactly one (combination of) correct update(s) on the base class(es) it stems from ([1]). Such views can be used to offer a certain filter on larger classes (or even combinations of classes) coming from some existing information system. Agents typically represent some part of an existing software system; they are equipped with suitable attributes, operations, and constraints on data and operations offering them appropriate functionality to negotiate with other agents about some activity, e.g. an activity pertaining to delegation of some service. In our treatment of delegation, we wish to abstract from certain details and concentrate on that kind of functionality that a delegator agent needs to (semantically) align with some other (supplier) agent. We propose to model delegator and supplier agents, in the context of semantic alignment, as exact views on top of existing information systems.

The next section deals with specifications of schemas for delegator agents and supplier agents. These schemas will rely heavily on employing exact views; these exact views will provide the mechanism to eventually construct so-called ω-maps, eventually ensuring correctness of alignment. We will also describe a running example of a company wishing to delegate its billing services.

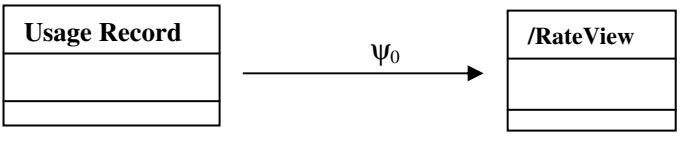
4 Delegation and Alignment by ω-Schemas and ω-Maps

Consider the case of our billing example, partially described in Section 1. To be more precise, the CSP wishes to delegate the calculation of the End Rates of the communication services used by its customers on the basis of: User Profile; Origination and destination of the communication session; Connection date; Connection start and stop time; Service type; Number of Bytes transported in case of a data service. The price a customer has to pay for its service usage, called *End Rate*, is based on two elements: the list price of the service (the so-called *basic rate*), and the subscription schemes a customer has agreed on with the CSP. Furthermore, the CSP wishes to obtain for each customer an invoice listing the rated sessions the customer has "consumed" the last period, say month, based on the rates of the Charge Records of that particular customer. In terms of a source agent, not all elements of the class Usage Record are relevant for outsourcing of these operations. To this end, we will construct a derived class with respect to the class Usage Record containing, in general, only those attributes, relations, operations, and constraints that are relevant to our particular delegation application. This particular derived class, called /RateView, will denote our source agent, and is depicted in the following figure.

/RateView
label: Set (subscriber, roamer) userId: String originationId: String destinationId: String connectionDate: Date connectionStartTime: Time connectionStopTime : Time serviceType: String volume: Integer
calcEndRate: Amount

(5)

A view such as /RateView is called a *source view*. In order to ensure a one-to-one correspondence between the view /RateView and the original class Usage Record (necessary to obtain a unique association between a usage-record object and his end rate), we provide a ψ-map ensuring that each object in the set of instances of class Usage Record corresponds to exactly one object in the view /RateView, and vice versa. In our example, we shall assume that there exists a ψ-map, say ψ_0, between Usage Record and /RateView, which we, using informal (i.e. non-UML) notation, depict below:



(6)

As mentioned in Section 1, we have the following σ-constraint, pertaining to the SLA that the delegating party demands from a prospective supplier.

```
context  Usage Record::calcEndRate: Amount
pre: Label = subscriber
post: Destination not in Europe implies result>2.50 € and
StartTime > 18:00 hrs implies result < 1.00 €
```

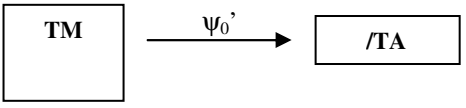
We shall now consider the side of the supplying party, and attempt to construct the target agent (or TA, for short). Should the company wish to delegate the operation `calcEndRate`, then the supplier is bound to the σ-constraint specified above. This σ-constraint entails that the supplier is to offer an implementation `calcEndRate'` of `calcEndRate`, such that `calcEndRate'` has a pre- and post condition that are in alignment with the pre- and post conditions of `calcEndRate`.

We assume that a target model TM is given, and that we are faced with the problem to define a view on TM, resulting in a target agent (TA), with the following properties:

(i) TA contains all relevant information also available in the source agent /RateView; we could also say that TA offers isomorphic copies of data structures available in SA.

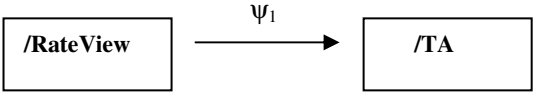
(ii) TA additionally contains from the target side extra attributes, operations, and relations (e.g., auxiliary tables with input data for operations) necessary to actually provide the calculations for the outsourced operations.

We therefore assume the existence of some ψ-map, say ψ_0', between TM and TA, as depicted below:



(7)

The view TA contains offers isomorphic copies of data structures available in /RateView; we shall assume that there exists a ψ-map, say ψ_1, between RateView and TA, as depicted in:

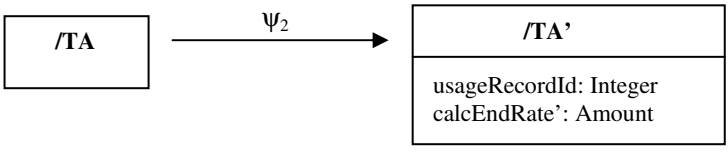


(8)

Existence of a ψ-map like ψ_1 is necessary for two reasons:

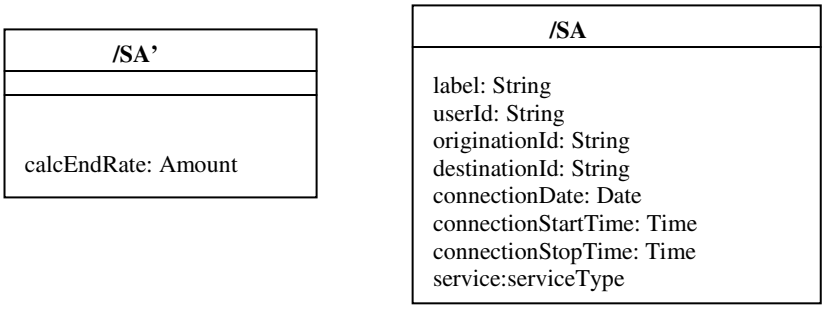
(i) ψ_1 maps object data from RateView to TA solving data extraction problems in the transition from the source model SM to the target model TM.
(ii) We have to ensure that each object in the set of instances of the source agent RateView corresponds to exactly one object in the target agent TA, and vice versa. Only in this way can we freely, and unambiguously move between the source and the target agents.

In order to provide sufficient material to implement `calcEndRate` as `calcEndRate'` in the view TA, we will employ auxiliary data found in classes in the target model TM. Examples of such auxiliary classes contain data, operations and constraints concerning list prices of service usage, taxes, promotions, credits, and debits. This wil result in yet another view, say TA', containing the following data.



(9)

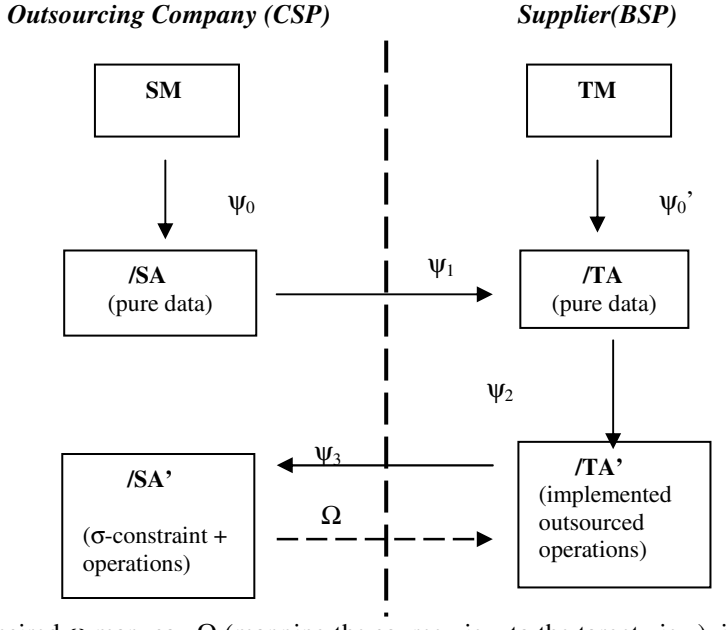
where we have assumed the existence of some ψ-map ψ_2 to map TA-objects to TA'-objects. TA' will also contain constraint information, defining the actual calculations of end rate. If we split the view `RateView` into two parts, one called SA containing purely the attributes, and the other called SA' containing purely the desired operations and the original σ-constraint, then we would have the following situation.



(10)

where SA' is a subclass of SA. By now providing a suitable ψ-map, say ψ_3, to map TA' to SA', we can now provide SA' with the desired calculations of calcEndRate.

We are now in the situation that we can construct our sought after ω-map as a composition of a certain sequence of ψ-maps. The diagram below offers an overview of the various ψ-maps encountered in our example thus far.



(11)

The desired ω-map, say Ω (mapping the source view to the target view), is defined as the inverse of the map ψ_3. Note that ψ-maps ψ_1 and ψ_2 are essential in defining Ω, since these two ψ-maps determine the target view `TV'`.

5 Conclusions and Future Work

We have shown how to use UML/OCL to model abstract versions of delegator agents and supplier agents in the context of semantic alignment. A formal means is provided to ensure that the delegation relationship between delegator and supplier, determined by a ruling service level agreement (SLA), satisfies correctness criteria defined in terms of consistency and completeness between the delegated operation and the associated operation offered by the supplier. Alignment correctness is ensured by satisfying so-called ω-schemas. Finally, negotiation between agents has been described as a strengthening and/or weakening of pre- and post conditions in the context of an ω-schema. Semantic alignment is, of course, only one aspect of actual alignment in a B2B setting. Business-related aspects (price, delivery time, reliability of the supplier, etc) and aspects pertaining to technical feasibility play an additional role. In this sense, a whole suite of agents play a role in reaching agreement on delegation of some service; cascading and nesting of agent responsibilities, for

example, are possible subjects of further research. Finally, we mention the research topic of the degree of automation support that can be reached in negotiations between two agents trying to agree on semantic alignment.

References

1. Balsters, H. ; Modeling Database Views with Derived Classes in the UML/OCL framework; «UML» 2003 6th Int. Conf.; LNCS 2863, Springer, 2003
2. Balsters, H., de Brock, E.O.; An object-oriented framework for reconciliation and extraction in heterogeneous data federations; Proc. 3rd Int. Conf. Advances in Information Systems, LNCS 3261, Springer, 2004
3. Balsters, H., de Brock, E.O.; Integration of integrity constraints in federated schemata based on tight constraining; Proc. OTM Confederated International Conferences CoopIS, DOA, and ODBASE , LNCS 3290, Springer, 2004
4. Bouzeghoub,M., Lenzerini, M; Introduction to: data extraction, cleaning, and reconciliation, Special issue; Information Systems 26 ; Elsevier Science, 2001
5. Dorfman, M., Thayer, R.H.; Software Engineering, Chapter 7, Software Validation, Verification, and Testing, Wiley, 1996
6. Eriksson, H.-E., M. Penker; Business Modeling with UML, Wiley, 2000
7. Franklin, S., Is it an Agent, or just a Program?:A Taxonomy for Autonomous Agents, Proc. 3rd Workshop Agent Theories, Architectures, and Languages, Springer, 1996.
8. Lenzerini, M.; Data integration; PODS'02, ACM Press 2002
9. Miller, R.J., Haas, L.M., Hernandez, M.A.; Schema mapping as query discovery; Proc. 26th VLDB Conf.; Morgan Kaufmann, 2000
10. Response to the UML 2.0 OCL RfP, Version 1.6, January 6, 2003
11. de la Rosa, J.L.,del Acebo, E., López, B. and Montaner, M., From Physical Agents to recommender agents*;* Intelligent Information Agents, Lecture Notes on Artificial Intelligence, Springer, 2002
12. Spaccapietra, S., Parent, C., Dupont, Y.; Model independent assertions for integration of heterogeneous schemas; VLDB Journal 1(1):81-126, 1992
13. Sparrow, E.; Successful IT Ousourcing, From Choosing a Provider to Managing the Project, Springer, 2003.
14. Szirbik, N.B.; A negotiation enabling agent-based infrastructure: composition and behavior; Information Systems Frontiers, Vol. 4, No. 1, 2002
15. Warmer, J.B., Kleppe, A.G.; The object constraint language; Addison Wesley, 2003
16. Wooldridge, M., Jennings, N., Intelligent Agents: Theory and Practice, Knowledge Engineering Review, vol. 10, no. 2, 1995

CAMS 2005 PC Co-chairs' Message

Context awareness is increasingly forming one of the key strategies for delivering effective information services in mobile contexts. The limited screen displays of many mobile devices mean that content must be carefully selected to match the user's needs and expectations, and context provides one powerful means of performing such tailoring. Context aware mobile systems will almost certainly become ubiquitous - already in the United Kingdom affordable 'smartphones' include GPS location support. With this hardware comes the opportunity for 'on-board' applications to use location data to provide new services - until recently such systems could only be created with complex and expensive components. Furthermore, the current 'mode' of the phone (e.g. silent, meeting, outdoors), contents of the built-in calendar, etc. can all used to provide a rich context for the user's immediate environment.

However, there is much to learn from a computer science perspective: context is a plastic and variable concept that can be realised in many ways - from the early notions of location-based services, through social navigation techniques based upon profiling of users, to concepts of work processes and information journeys. Together, these differing forms of context provide a challenging diversity of data which needs to be brought together and consistently and rapidly processed. These demands provide a strong testbed of contemporary techniques for modelling context, particularly when the network and processing capacities of mobile systems are considered.

At this year's, first, Context Aware Mobile Systems (CAMS) workshop, we have a strong set of paper presentations spread over two days. We are sure that workshop attendees will be delighted with the breadth and depth of contributions that are to be discussed. Papers cover the spectrum of context-aware mobile systems: the traditional basis of location, the processes of personalisation and profiling, emerging areas such as context-aware querying and engineering requirements such as development models and architectural frameworks. The global nature of the research in this area is also reflected in the wide spread of countries represented by the paper authors.

August 2005

Annika Hinze, University of Waikato
George Buchanan, University College London
(CAMS'05 Program Committee Co-Chairs)

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, p. 121, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Personalising Context-Aware Applications*

Karen Henricksen[1] and Jadwiga Indulska[2]

[1] CRC for Enterprise Distributed Systems Technology (DSTC)
karen@itee.uq.edu.au
[2] School of Information Technology and Electrical Engineering,
The University of Queensland
jaga@itee.uq.edu.au

Abstract. The immaturity of the field of context-aware computing means that little is known about how to incorporate appropriate personalisation mechanisms into context-aware applications. One of the main challenges is how to elicit and represent complex, context-dependent requirements, and then use the resulting representations within context-aware applications to support decision-making processes. In this paper, we characterise several approaches to personalisation of context-aware applications and introduce our research on personalisation using a novel preference model.

1 Introduction

Context-awareness has emerged as a popular design approach for building adaptive applications for mobile and pervasive computing environments. Context-aware applications rely on information about the context of use - such as the user's current location and activity - to provide seamless operation in the face of mobility and intelligent support for users' evolving requirements.

As users of context-aware applications can differ greatly in terms of their requirements and expectations about how their applications should behave, personalisation mechanisms are required. Unfortunately, personalisation of context-aware applications is substantially more challenging than personalisation of traditional desktop applications. Because the actions of context-aware applications are partially determined by the context, user preferences must likewise be predicated on context. The set of distinct contexts recognised by a context-aware application may be large, implying that the set of user preferences might also be large and complex. A further problem related to personalisation is the need to provide users with a clear mental model and appropriate feedback mechanisms that allow them to understand the links between application behaviours and their specified preferences. These are essential in order to prevent user frustration at apparently erratic behaviour, and to facilitate trouble-shooting.

* The work reported in this paper has been funded in part by the Co-operative Research Centre for Enterprise Distributed Systems Technology (DSTC) through the Australian Federal Government's CRC Programme (Department of Education, Science, and Training).

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 122–131, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Owing to the immaturity of the field of context-aware computing, very little research has addressed personalisation. Rather, the main focus has been on techniques for acquiring, interpreting and managing context information from sensors. In this paper, we survey the limited work that has been done in the area and then report on our own research, which has investigated personalisation based on a novel preference model.

The structure of the paper is as follows. Section 2 provides an analysis of three approaches that have been pursued for personalisation of context-aware applications, while Sections 3 and 4 introduce the preference model used in our work. Section 5 briefly touches on implementation issues, describing a software infrastructure and programming toolkit we developed to facilitate the use of our preference model by context-aware applications. Finally, Section 6 highlights some user interface design issues and Section 7 discusses topics for future work.

2 Personalisation Approaches

The existing approaches to personalising context-aware applications can be classified as follows:

- *End-user programming approach.* This approach offers the most radical form of personalisation: rather than applications being developed by software engineers with hooks for customisation by users, end-user programming techniques place the task of constructing applications into the hands of users. Several styles of end-user programming exist, including programming by demonstration [1], in which users train a system to carry out desired actions by manually demonstrating the actions, and programming by specification [2], in which users provide high-level descriptions of desired actions.
- *User modelling/machine learning approach.* This approach removes responsibility for personalisation from the user, instead using machine learning techniques to automatically derive user requirements from historical data. These requirements can be represented in the form of user models suitable for use by applications as a basis for adaptive and pro-active behaviours [3].
- *Preference-based approach.* This approach is the closest to traditional personalisation approaches for desktop applications. It typically relies on user interfaces or configuration files through which users can manipulate settings or rules that control the way applications react to context. Other than our own work, we are not aware of any research that addresses the general problem of personalising context-aware applications with context-dependent preferences; however, several applications that provide custom-designed preference mechanisms have been developed (e.g., [4]).

Each approach has shortcomings which limit its applicability to certain application domains. The main problem of end-user programming techniques is that they are generally suitable only when the application behaviours that users need to specify are reasonably simple. For complex tasks, users experience difficulties in demonstrating or specifying their requirements. As a result, the most common

scenarios presented in the literature on end-user programming concentrate on simple tasks, such as loading presentation files in advance of a meeting [1]. A further problem is that most of the end-user programming solutions are primarily concerned with supporting the initial programming task, and it is unclear how well they can support evolution as user requirements or the environment change. Finally, a large initial investment is expected of users to either train the system or specify the required behaviours.

The second approach, based on user modelling and machine learning, is more appropriate than end-user programming for complex applications and does not require a period of explicit training or set-up by the user. However, mistakes are nearly always made during the learning process, causing frustration to the user. The user can provide feedback to help prevent similar mistakes in the future; however, many rounds of feedback may be required before the desired behaviour emerges. Users may prefer to avoid the frustration of repeatedly providing feedback by explicitly specifying some or all of their requirements; however, manual customisation is unfortunately not supported in this approach.

The preference-based approach does not suffer from this problem, as it allows users to explicitly specify requirements at any time. It can also be used in conjunction with automated preference learning mechanisms, so as to reduce the burden on users to specify preference information that is complete and up to date. Finally, unlike end-user programming, the preference-based approach is appropriate even for complex applications. This means that it is arguably the most promising and widely applicable of the three approaches.

3 A Preference Model for Context-Aware Applications

The remainder of the paper focuses on our preference-based personalisation approach, which is based on a novel preference model. This section introduces the model, while preference examples are deferred until following section.

When starting our work on personalisation, we surveyed preference modelling approaches from diverse fields such as decision theory and document retrieval, with the aim of identifying a preference model that could be used as a basis for personalisation of context-aware applications. This survey can be found in [5]. However, none of the approaches that we examined was able to represent context-dependent preferences. Accordingly, we developed our own preference model designed to address this limitation. This model supports user-customisable decision-making by context-aware applications, as shown in Fig. 1.

In this decision-making process, user preferences are evaluated against a set of context information, candidate choices (which may be associated with one or more corresponding actions) and application state variables, to yield an assignment of ratings to the candidate choices. The user preferences may reflect the requirements of one or multiple users. Arbitrary kinds of choices can be supported: for example, the choices may be documents or search terms in the case of an information retrieval application, or email folders in the case of an email filtering tool.

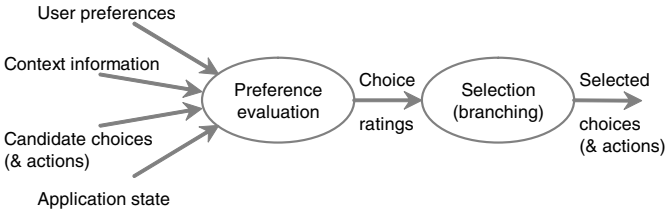


Fig. 1. Context- and preference-based decision process for context-aware applications

After using the preferences to rate the candidate choices, the context-aware application selects zero or more of the choices, and carries out a set of corresponding actions (for example, displaying a set of chosen documents to the user or filtering an email to a selected folder). We refer to this step as *branching*.

Each user preference takes the form of a scope and a scoring expression. The scope specifies the context and choices to which the preference applies using a special form of logical predicate, which we term a *situation*. Our notation for defining situations is described in an earlier paper [6], in which we outlined our approach to context modelling, and therefore will not be described in detail here. The scoring expression produces a rating that indicates the suitability of the choices that match the scope within the given context. This rating is one of the following:

- a numerical value in the range [0,1], where increasing scores represent increasing desirability;
- *prohibit*, indicating that a choice must not be selected in a given context;
- *oblige*, indicating that a choice must be selected in a given context;
- *indifferent*, indicating an absence of preference; or
- *undefined*, signalling an error condition (e.g., an attempt to combine *prohibit* and *oblige* scores).

Preferences may be either *simple* preferences, which express atomic user requirements, or *composite* preferences, which specify how other preferences are combined to produce appropriate aggregate ratings. Related preferences can also be dynamically grouped into preference sets. In the following section, we present some example preferences and preference sets.

4 Preference Examples

Although user preferences may be generic enough to be applied to many context-aware applications, most preferences are specific to a particular application (and type of choice within that application). Here, for illustrative purposes, we focus on the specification of preferences for a context-aware email client. This application uses context information to enhance a set of standard email management features, such as automatic forwarding and filtering of messages, alerting for important messages, and auto-replying to messages. The use of context-awareness

allows the email client to behave more pro-actively than would otherwise be possible. For instance, the client can automatically produce auto-reply messages when the user is on vacation (without any prior set-up) and perform message filtering on arbitrary types of context, not only on message headers and content.

In Table 1, we show some sample user preferences related to filtering. The goal of filtering is to assist with organising email into folders. These preferences assume that the user's email folders include the following: inbox, inbox-secondary, personal and level7. Four simple preferences (*l7_unread*, *l7_read*, *personal* and *important*), one composite preference (*filtering*), and one preference set (*filtering_set*) are shown.

The scope of each preference follows the "when" keyword, while the scoring expression is preceded by the "rate" keyword. The first preference states that filtering to the secondary inbox is highly preferred when:

- the message is addressed to the level7 mailing list ("level7@dstc.edu.au");
- the user is currently out of the office (which possibly implies that the message is irrelevant, as much of the list traffic is solely of interest to the current occupants of floor level 7); and
- the message is new (i.e., unread).

The second preference states that already read messages addressed to the same mailing list should be filed in the level7 folder, with high preference. The *personal* preference states that messages from family members or friends should be filed to the personal folder, with medium-high preference, while the *important* preference states that unread messages that have the highest priority level must

Table 1. Example preferences for email forwarding

l7_unread =	`when` *outOfOffice*(*me*) and *contains*(*to*, "level7@dstc.edu.au") and *equals*(*status*, "new") and *equals*(*folder*, "inbox-secondary") `rate` *high*
l7_read =	`when` *equals*(*status*, "read") and *contains*(*to*, "level7@dstc.edu.au") and *equals*(*folder*, "level7") `rate` *high*
personal =	`when` (*familyMembers*(*me*, *sender*) or *friends*(*me*, *sender*)) and *equals*(*folder*, "personal") `rate` *medium_high*
important =	`when` *equals*(*priority*, "highest") and *equals*(*status*, "new") and *equals*(*folder*, "inbox") `rate` *oblige*
filtering_set =	{*l7-unread*, *l7-read*, *personal*, *important*}
filtering =	`when` true `rate` *average*(*filtering_set*)

always remain in the user's inbox. The preference ratings *medium_high* and *high* are mapped to numerical values as defined by the application developer, in order to allow aggregation of scores. Note that, although the preferences are somewhat similar in appearance to the user-defined message filters that are already supported by email clients such as Mozilla Thunderbird and Microsoft Outlook, two of the preferences (*l7_unread* and *personal*) refer to external context definitions (i.e., the *outOfOffice*, *familyMembers* and *friends* situations) that cannot be included in standard email filters.

To combine the requirements expressed by these four simple preferences to support decision making about how to filter messages, the preferences are first grouped into a preference set (*filtering_set*). The *filtering* composite preference then defines the overall rating for a given folder as the average of the ratings produced by the preferences in this set. Here, averaging is performed according to the following simple algorithm:

1. if any preference produces the *undefined* score, the result of averaging is the *undefined* score; else
2. if one or more preferences produces the *oblige* score *and* one or more preference produces the *prohibit* score, then the result is the *undefined* score; else
3. if one or more preferences produces the *oblige* score, then the result is the *oblige* score; else
4. if one or more preferences produces the *prohibit* score, then the result is the *prohibit* score; else
5. if one or more preferences produces a numerical score, then the result is the average of all numerical scores; else
6. if all preferences produce *indifferent* scores (which occurs by default when the preference scopes do not hold), then the result is the *indifferent* score.

To illustrate, we consider the scenario in which the email client filters an already read message that was sent by a friend of the user to the level7 mailing list. The ratings produced by the preferences defined in Table 1 are shown in Table 2. Only the *l7_read* and *personal* preferences are relevant to this example. The remaining preferences produce indifferent ratings for all four email folders. As the *l7_read* preference produces a higher preference rating, this preference takes precedence. Therefore, the message in this example would be filtered to the level7 folder, rather than the personal folder.

Table 2. Preference ratings for an already read message sent by a friend to the level7 mailing list

Preference	**inbox**	**inbox-secondary**	**personal**	**level7**
l7_unread	*indifferent*	*indifferent*	*indifferent*	*indifferent*
l7_read	*indifferent*	*indifferent*	*indifferent*	*high*
personal	*indifferent*	*indifferent*	*medium-high*	*indifferent*
important	*indifferent*	*indifferent*	*indifferent*	*indifferent*
filtering	*indifferent*	*indifferent*	*medium-high*	*high*

5 Infrastructural Support for Personalisation

To assist with implementing context-aware applications that support personalisation using our preference model, we have developed a layered software infrastructure that supports:

- integration, management and querying of context information from various sources, including sensors, context-aware applications and human users (*context management layer*);
- management and evaluation of user preference information (*preference management layer*); and
- decision making and branching at the application layer, using the services of the context and preference management layers (*programming toolkit*).

The context and preference management layers are implemented in Java, using relational databases for information storage and management. However, they accept requests via several different communication protocols (XML/HTTP, Java RMI and Elvin [7]), and therefore can be used in conjunction with a variety of platforms and programming languages. The programming toolkit, which provides various helper classes for formulating decision problems and selecting appropriate actions based on the ratings produced by the preference management layer, can currently only be used by Java applications, but could be ported to other languages in the future. Further information about the software infrastructure can be found in some of our earlier papers [6,8].

6 User Interface Design

The preference notation that we described in Sections 3 and 4 is used internally by our programming toolkit and preference management layer, but would rarely be exposed directly to users. In this section, we discuss some issues related to the design of user interfaces to support personalisation of context-aware applications. Appropriate user interface designs must necessarily be considered on a case-by-case basis; because of this, our discussion focuses on the design of a user interface for the email application we discussed in Section 4, as a case study. However, we also offer a set of general design guidelines in Section 6.2.

6.1 Personalisation Interfaces for Context-Aware Email

Email applications provide a useful starting point for thinking about user interface design issues, as most already support personalisation. Therefore, instead of thinking about how personalisation can be incorporated from scratch, it is only necessary to think about how to extend the existing personalisation to support context-dependent user preferences. We have been working on a set of context-aware extensions for the Thunderbird email client[1].

[1] http://www.mozilla.org/products/thunderbird/

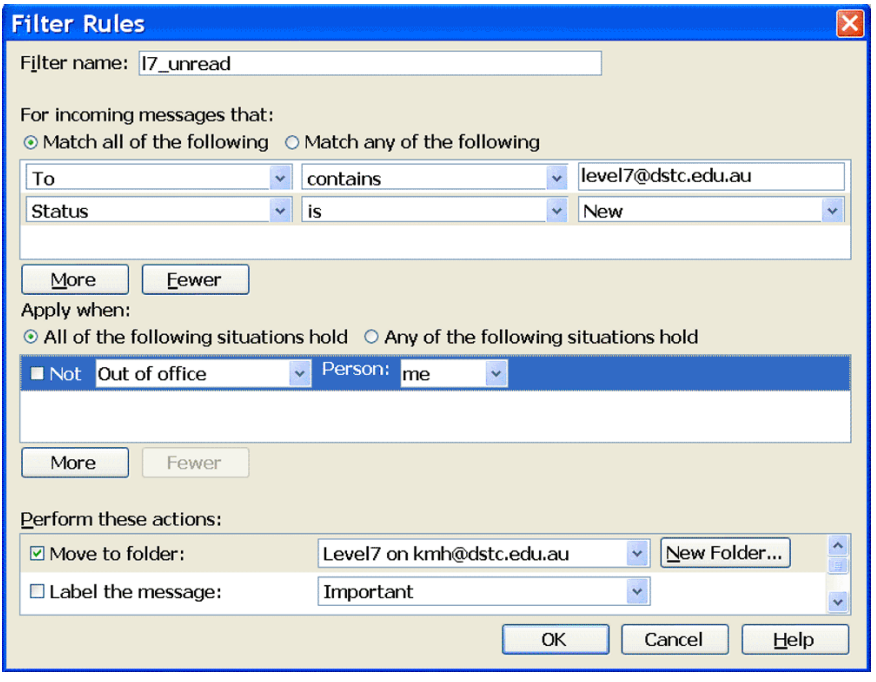


Fig. 2. An extension of the current Thunderbird filter rule interface to support context-dependent filters

Thunderbird supports many types of personalisation, but here we focus on filtering. Users can define new message filters, and enable/disable already created filters. Filters are specified as matching conditions, defined in terms of message headers, and corresponding actions (e.g., "move a message to a specified folder"). To support context-based filtering, only minor changes are needed to the existing user interface. In Fig. 2, we show a trivial extension of the current interface which allows the matching conditions to be augmented with relevant situations. Each filter can be mapped to a preference that conforms to the preference model described in Sections 3 and 4. Observe, for example, that the filter defined in Fig. 2 matches the scope of the *l7_unread* preference in Table 1. The rating assigned to this preference/filter is set on another screen (not shown), which lists all defined filters and provides controls for enabling/disabling filters.

The fact that our preference model provides such a close fit with the existing personalisation model used by Thunderbird - despite the fact that we did not have email in mind as a target application when designing the preference model - helps to validate the design of the model. As we show in this example, personalisation interfaces should be consistent with the general appearance and function of a context-aware application, rather than being closely tied to the preference model. In particular, a personalisation interface should always be more than a simple preference editor that expects the user to directly formulate preferences of the kind shown in Table 1.

6.2 General Design Guidelines

To date, we have built a number of personalisable context-aware applications using our preference model. These have included several communications applications [6,9], a vertical handover application for managing the streaming of multimedia to mobile users [8], and virtual community applications to support independent living of the elderly [10]. As a result of our experiences with these applications, we offer the following general design guidelines:

- *Constrain the types of personalisation that can be performed by users.* That is, even when using a generic preference model such as the one we have presented, the full power of the model should not be exposed to users. There are two reasons for this: (i) users are likely to be overwhelmed and confused, and (ii) complex preference sets should be thoroughly tested before they are deployed to ensure that no unexpected behaviours emerge.
- *Integrate personalisation mechanisms into the everyday use of the application.* This helps to ensure that personalisation is a natural and visible part of the application, and increases the chance that users will use and understand the personalisation mechanisms. This design principle is somewhat similar to the one advocated by Lederer et al. [11] in relation to designing systems to support privacy.
- *Provide logging and feedback mechanisms.* These should let users (i) see how their preferences are linked to actions and (ii) override actions if necessary. Logging and feedback can help to prevent user frustration and assist users with correcting preference and/or context information when required.
- *Provide useful default behaviours.* That is, ensure that most people will be able to use the application reasonably well from first use, even without any personalisation. Some people will resist using personalisation mechanisms at all, no matter how visible and straightforward they are.

7 Future Work

This paper outlined our efforts to develop a preference model for personalisation of context-aware applications. As discussed in Section 6.2, we have used the model in conjunction with a variety of context-aware applications. Although our experiences with using the model have been positive, we have already identified some important refinements and extensions for future work. In particular, we have started designing some modifications to the preference model that should improve both the usability of the model for application developers and the efficiency of preference evaluation. We have also begun working on techniques for automated preference learning based on user feedback. In the near future, we hope to extend our programming toolkit and preference management system to support these mechanisms. In the longer term, we plan to investigate extensions of the preference model to a broader set of decision problems relevant to context-aware applications. At present, our model is best suited to choices over a fixed (and reasonably small) set of alternatives; in the future, we plan to study decision problems that are both larger and more open-ended. Finally, appropriate

user evaluation is crucial, not only for our preference model, but also for the other personalisation approaches discussed in Section 2.

References

1. Dey, A.K., Hamid, R., Beckmann, C., Li, I., Hsu, D.: a CAPpella: Programming by demonstration of context-aware applications. In: ACM Conference on Human Factors in Computing Systems (CHI), Vienna (2004)
2. Truong, K.N., Huang, E.M., Abowd, G.D.: CAMP: A magnetic poetry interface for end-user programming of capture applications for the home. In: 6th International Conference on Ubiquitous Computing (UbiComp). Volume 3205 of Lecture Notes in Computer Science., Springer (2004) 143-160
3. Byun, H.E., Cheverst, K.: Harnessing context to support proactive behaviours. In: ECAI2002 Workshop on AI in Mobile Systems, Lyon (2002)
4. Lei, H., Ranganathan, A.: Context-aware unified communication. In: 5th International Conference on Mobile Data Management (MDM), Berkeley (2004)
5. Henricksen, K.: A Framework for Context-Aware Pervasive Computing Applications. PhD thesis, School of Information Technology and Electrical Engineering, The University of Queensland (2003)
6. Henricksen, K., Indulska, J.: A software engineering framework for context-aware pervasive computing. In: 2nd IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE Computer Society (2004) 77-86
7. Segall, B., Arnold, D., Boot, J., Henderson, M., Phelps, T.: Content based routing with Elvin4. In: AUUG2K Conference, Canberra (2000)
8. Henricksen, K., Indulska, J., McFadden, T., Balasubramaniam, S.: Middleware for distributed context-aware systems. International Symposium on Distributed Objects and Applications (DOA) (to appear) (2005)
9. McFadden, T., Henricksen, K., Indulska, J., Mascaro, P.: Applying a disciplined approach to the development of a context-aware communication application. In: 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE Computer Society (2005) 300-306
10. Indulska, J., Henricksen, K., McFadden, T., Mascaro, P.: Towards a common context model for virtual community applications. In: 2nd International Conference on Smart Homes and Health Telematics (ICOST). Volume 14 of Assistive Technology Research Series., IOS Press (2004) 154-161
11. Lederer, S., Hong, J.I., Dey, A.K., Landay, J.A.: Personal privacy through understanding and action: five pitfalls for designers. Personal and Ubiquitous Computing 8 (2004) 440-454

Management of Heterogeneous Profiles in Context-Aware Adaptive Information System

Roberto De Virgilio and Riccardo Torlone

Dipartimento di Informatica e Automazione,
Università degli studi Roma Tre
{devirgilio, torlone}@dia.uniroma3.it

Abstract. Context-awareness is a fundamental aspect of the ubiquitous computing paradigm. In this framework, a relevant problem that has received little attention is the large heterogeneity of formats used to express a context information: text files in ad-hoc format, HTTP header, XML files over specific DTD's, RDF, CC/PP and so on. So many applications meet difficulties in interpreting and integrating context information coming from different sources. In this paper we propose an approach to this problem. We first present a general architecture for context-aware adaptation that is able to take into account different coordinates of adaptation. We then show how, in this framework, external profiles are dynamically captured and translated into a uniform common representation that is used by the system to meet the requirements of adaptation. We also present a prototype application that implements the proposed approach.

1 Introduction

A today typical scenario of a Web based Information System is the following: in response to a generic request done by a user with his/her portable device the system (i) automatically generates a hypermedia presentation that better meets the user needs and (ii) delivers the result in a format that is suitable for the access device. It follows that a novel and fundamental requirement of modern Web Information Systems is the ability to personalize and adapt content delivery according to the *context* (or *profile*) of the client. In the literature, several definitions of context can be found. In general, the term is adopted to indicate "a set of attributes that characterizes the capabilities of the access mechanism, the preferences of the user and other aspects of the context into which a Web page is to be delivered" [17]. These may include the access device, the network QoS, the user preferences, the location, and so on.

It is widely recognized that a well designed context model is a fundamental ingredient of any context-aware system and actually a variety of context models have been proposed in recent years [3,12,14,15]. The objective of most research on this topic is the development of uniform context models and query languages, as well as reasoning algorithms, to facilitate the representation and management of contexts and the interoperability of applications.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 132–141, 2005.
© Springer-Verlag Berlin Heidelberg 2005

The various context modeling approaches can be classified according to the data structures which are used to exchange contextual information in the respective systems. Actually, the most simple data structure used is based on attribute-value pairs.

It is a common practice to express context models by means of markup languages, characterized markup tags that identify attributes and content organized into a hierarchical data structure. These approaches require a serialization of context information in XML, the most popular mark-up langauge. Some of them are based on the Composite Capabilities / Preferences Profile (CC/PP) [18] and User Agent Profile (UAProf) [19] standards, which have the expressiveness provided by RDF, a language for representing (meta) information about resources in the World Wide Web. These kinds of context modeling approaches usually extend and complete the basic CC/PP and UAProf vocabulary and procedures to try to cover the higher dynamics and complexity of contextual information compared to static profiles. An example of this approach is the Comprehensive Structured Context Profile (CSCP) proposed by Held [8].

In this framework, in spite of many proposed approaches to the problem of adaptation of Web Information Systems [1,2,7,11,13,5,6,16], a relevant problem that has received little attention is how to manage in a coordinate way the large heterogeneity of formats used to express a context information: text files in ad-hoc format, HTTP headers, XML files over specific DTDs, RDF, CC/PP and their dialects.

In this paper we propose a general notion of profile that can be used to represent a large variety of contexts. We then present a translation methodology that takes as input generic context information, possibly expressed in different formats, and generates a representation in such a general model. This methodology is implemented in a component of a general and flexible architecture for content adaptation based on the management of profiles [4]. In this architecture, the analysis of profiles drives the generation of a configuration that specifies, at the various layers of a Web based Information System (content, navigation and presentation), how to build a response that meets the requirements of adaptation of the profile. We also describe architecture and functionality of a prototype implementing such translation module and illustrate practical examples of translations.

The paper is structured as follows. Section 2 briefly illustrates the general architecture of our adaptation system. Section 3 describes our context model and presents an interpreter that maps heterogeneous contexts into this model. Section 4 presents a practical implementation of the interpreter based on an extension of CC/PP format. Finally, in Section 5 we sketch some conclusions and future work.

2 A General Architecture of an Adaptation System

In Web-based Information Systems (WIS), it is very useful to consider separately the three main components: the *content* (that is, the data to publish),

the *presentation* (that is, the layout of the pages where to publish selected data) and the *navigation* (that is, the hypertext structure of web site). An adaptation process should operate on all these components by: selecting the most appropriate content (e.g., according to user interests), building an adequate layout for the Web pages (e.g., according to layout capabilities of the client device) and organizing the hypertext structure of the web interface (e.g., decomposing large contents in linked pages, when the band of the communication channel is limited).

A general architecture of a system able to meet these requirements is the one reported in Figure 1.a.

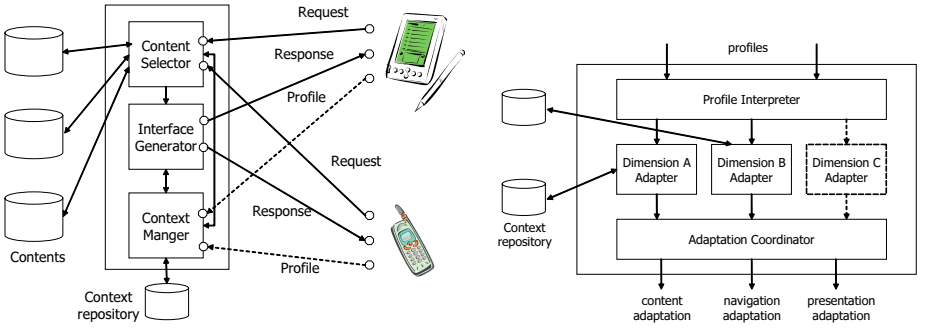


Fig. 1. A general architecture of reference

This includes:

- a Request Interpreter (RI), able to translate a specific user request (a page or a specific object) into a query over the underlying data;
- a Response Generator (RG), able to generate all the components of a response to deliver over the Web (that is, content, structure and layout) that satisfies the given request and is appropriate for the client profile;
- a Context Manager (CM), able to get and manage a description of the client characteristics (usually called profile), and support the Response Generator in the execution of its task.

In this architecture, a fundamental role is covered by the Context Manager that should be able to:

1. (dynamically) capture (possibly heterogeneous) incoming contexts of clients and translate them into a uniform (and general) representation,
2. coordinate the various requirements of adaptation for a given context,
3. send to the Response Generator some *adaptation specifications* that can be used to build the response at the various levels (content, navigation and presentation).

To guarantee the flexibility of the overall system, this component should be extensible, in the sense that the various activities should be carried out for different types of context and according to orthogonal dimensions of adaptation, possibly not fixed in advance.

In Figure 1.b it is reported a possible architecture for the Context Manager that can meet these requirements. The basic component is the Profile Interpreter, which should be able to get and identify possibly heterogeneous contexts (e.g., expressed in CC/PP, XML, HTTP header) and translate them into a uniform representation. Such representations are taken as input by a series of modules, one for each dimension of adaptation (e.g., the device characteristics, the user preferences, the location, etc.). The main task of these modules is to generate a uniform set of adaptation specifications, to be sent to the Response Generator, that satisfy the specific requirements of one dimension. This work can be supported by a specific data repository in which predefined or previously generated contexts and corresponding specifications are collected. Since each module can generate different and possibly conflicting specifications, a coordination is needed to provide an integrated set of specifications that take into account the various adaptation requirements and can be effectively sent to the RG module. The Adaptation Coordinator is devoted to the execution of this task.

It is important to note that, due to the uniformity of representations and techniques used by the various adaptation modules, this scheme can be extended in a natural way: a new adaptation module can be easily added to satisfy the requirements of adaptation according to a previously unpredicted coordinate.

In [4] we have proposed a methodology that leads the various activities of the Context Manager. In the rest of this paper, we will focus our attention to the Profile Interpreter and present methods and tools for the management of heterogeneous profiles.

3 Profile Interpretation

In this section we present GPM, a general model of context, and illustrate a mechanism for context interpretation that is based on this model.

3.1 General Profile Model

In GPM (a short hand for *General Profile Model*), a profile is a description of an autonomous aspect of the context in which the Web site is accessed that should influence the delivery of its contents. Examples of profiles are the user, the device, the location, and so on. A *dimension* is property that characterizes a profile. Each dimension can be conveniently described by means of a set of *attributes*. Each attribute can be *simple* or *complex*. A simple attribute has a value associated with it, whereas a complex attribute has associated a set of (simple or complex) attributes.

For example, a profile for a client device can be represented by means of the hardware and software dimensions. In turn, the hardware dimension can be described by means of a simple attribute like CPU and a complex attribute like display, composed by the simple attributes width and height. In GPM, a *context* is just a collection of profiles. Examples of profiles for the context of a client A are reported in Figure 2.

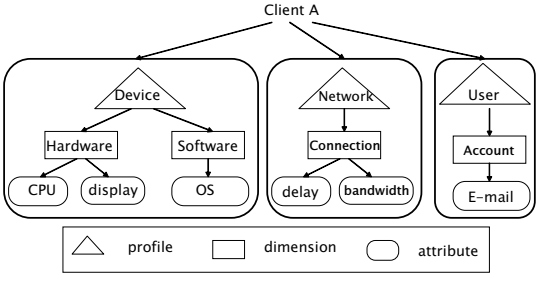


Fig. 2. Example of profiles for a particular context

The instance I_A of an attribute A is defined as follows. If A is simple, I_A is a pair (A, v), where v is a value. Otherwise, I_A is a set of pairs (A, I_{A_k}), for each attribute A_k that composes A, where I_{A_k} is an instance of A_k. The instance I_D of a dimension D is a set of pairs (D, I_A), where I_A is an attribute instance, for each attribute A of D. Finally the instance of a profile P is a set of dimension instances, for each dimension of P.

Note that this notion of profile is very general and is therefore suited to model almost all profile formalisms proposed in the literature and adopted in practical systems.

3.2 The Interpretation Method

Our adaptation methodology operates over profiles that have to be known in advance so that the adaptation engine can extract from them relevant data to produce suitable adaptation specifications [4]. Therefore, given a source instance PI_s of a profile PS_s, described according to a context model PM_s, we need to generate a target instance PI_t of a given profile PS_t in the GPM model, containing the same information as PI_s.

As indicated in Figure 3, this is done in two steps. First, PI_s (and PS_s) is translated in the GPM model and then it is *transformed* to generate PI_t. Given the generality of the GPM model, the first step is rather easy since it mainly requires a rewriting activity. The second step is more involved and is based on a *mapping* between *profiles* that drives the profile instance PI_t .

Definition 1 (Mapping of profiles). *Given two profiles P_1 and P_2 in GPM model, a mapping M is a set of pairs (s, t) where s is a component (dimension or attribute) of P_1 and t is a component of P_2.*

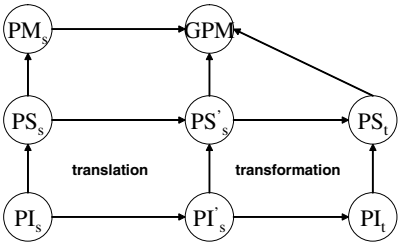


Fig. 3. Translation process

3.3 The Transformation Algorithm

In Figure 4 is reported an algorithm that takes as input a profile instance PI_s of a profile PS_s, a profile PS_t, and a mapping M between PS_s and PS_t, and returns an instance PI_t of PS_t.

Algorithm 1
Input: *PI_s, PS_s, M, PS_t*
Output: *PI_t*
(1) ***begin***
(2) *$\forall$ dimension $D \in PS_t$*
(3) *if $\exists m \in M / m = (D',D)$*
(4) *$\forall$ attribute A of $D \in PS_t$*
(5) *if $\exists m' \in M / m' = (A',A)$*
(6) *$PI_t \cup (D,VALUE(A,m'))$;*
(7) ***end***

Procedure 1
(1a) I_A VALUE(attribute A, pair m) {
(2a) ***if*** *(A is simple)* ***return*** *(A,value of m.s);*
(3a) ***else*** *{*
(4a) *I_A vA = $\varnothing$;*
(5α) *$\forall$ attribute A' of A {*
(6a) *if $\exists m' \in M / m' = (A'',A')$*
(7a) *vA $\cup$ (A', VALUE(A',m'));*
(8a) *}*
(9a) ***return*** *vA;*
(11a) *}*
(12a) }

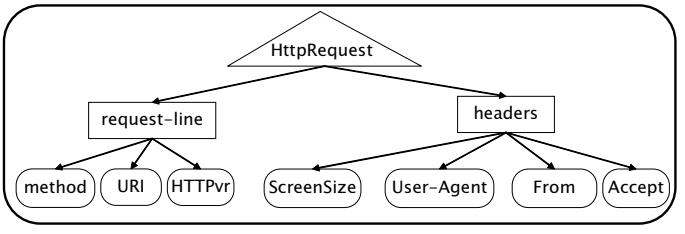
Fig. 4. Example of profiles for a particular context

The algorithm iterates over the components of PS_t. For each dimension D (line (3)), if there is a pair $(D', D) \in M$, for each attribute A of D, it searches a pair $(A', A) \in M$. If the pair (A', A) exists, then an instance of A is generated. This is done by invoking an internal procedure. This procedure proceeds as follows. If A is simple, it returns the instance of A'. Otherwise, the procedure recursively considers each attribute A'' of A.

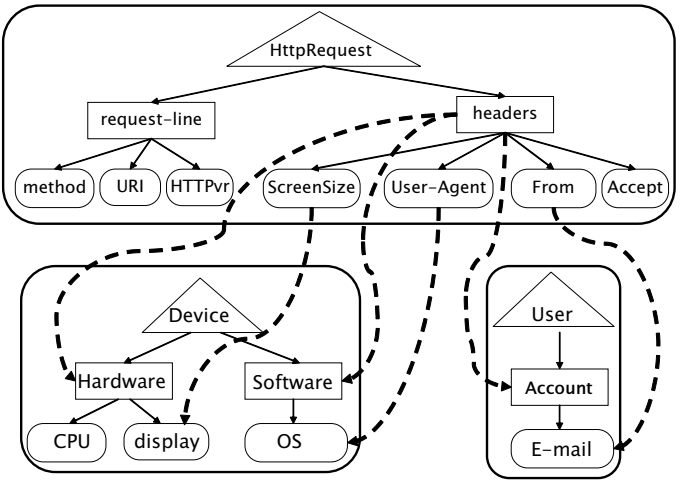
In figure 5 we show an example of whole interpretation process. Let us assume that the input context C is represented by means of an HTTP header (Figure 5.a). First, C is translated into a context C' in the GPM model (figure 5.b). Then, we define a mapping between the profiles included in C' and the target profiles (Figure 5.c). By applying the transformation Algorithm we obtain the GPM context reported in Figure 5.d.

```
POST register.jsp HTTP/1.1
Host: hi.iq
User-Agent: Mozilla/5.0
Accept: text/xml, application/xml, application/xhtml+xml,
        text/html, text/plain, video/xmng, image/png,
        image/jpeg, image/gif, text/css, */*
From: w3c@yahoo.com
ScreenSize: 15x10
```

(a)



(b)



(c)

{(Device, [(Hardware, (Display,15x10)),(Software,(OS,Mozilla/5.0))]),
(User, [(Account,(E-mail,w3c@yahoo.com))])}

(d)

Fig. 5. An example of profile interpretation

4 A Practical Implementation

4.1 A CC/PP Extension

A popular format used to express profiles is CC/PP. Indeed, a CC/PP profile is limited to a two-level hierarchy (component–attribute). In the same line of other approaches [8,10], we have extended CC/PP to express GPM profiles. So each component of this formalism represents a dimension of the profile and each attribute can be simple or complex.

A set of wrappers are used to translate external profiles into such a CC/PP extended format. In our approach, the resulting profiles are then mapped to a set of target profiles that are taken as input by the adaptation process. The profile interpreter able to execute these activities has a scheme reported in Figure 6.

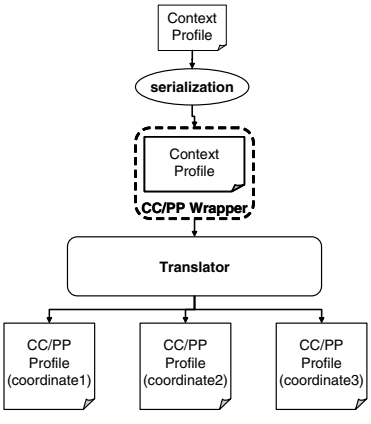


Fig. 6. The Profile Interpreter

It includes two main components:

- a **translator** that wraps an external context and serializes it into our extended CC/PP format;
- a **transformer** that defines mappings between the profiles returned by the translator and the internal profiles used by the Context Manager.

4.2 A Prototype Application

We have designed and developed a prototype implementation of the Profile Interpreter. This tool makes use of Jena, a Java framework for building RDF based applications [9] and is equipped with several wrappers (both for profiles based on an attribute-value model and for profiles based on markup languages). The main features of the tool are the following:

- if needed, it generates XML representations of the schema of an external profile;
- it supports the user in the definition of mappings between profiles by means of a user-friendly interface (Figure 7);
- it interprets profiles using the technique presented in the previous section.

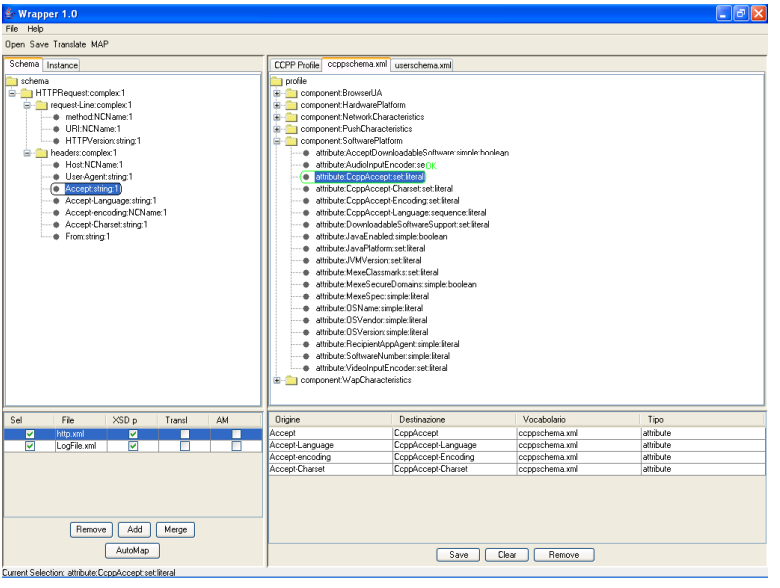


Fig. 7. The prototype application

5 Conclusions and Future Work

In this paper, we have presented an approach to the problem of the management of heterogenous representation of profiles (text files in ad-hoc format, HTTP headers, XML files over specific DTDs, RDF, CC/P) in context-aware information systems. The approach is based on a general notion of profile that can be used to represent a variety of contexts at different level of details. A translation technique is used to translate external profiles into such a uniform common representation. This functionality is embedded into a general architecture for content adaptation.

From a conceptual point of view, we are currently investigating in more depth the notions of profile and mapping, in order to improve their generality and usability. From a practical point of view we are extending the features of the context manager, in particular by enhancing the profile interpretation capabilities. This will be done by deriving in an automatic way potential mappings between profiles, making use of external information, for instance in the form of ontologies.

References

1. T. Bickmore, A. Girgensohn, and J. Sullivan. Web page filtering and reauthoring for mobile users. *Computer Journal* , 42(6):534–546, 1999.
2. S. Ceri , P. Fraternali, A. Bongio , M. Brambilla, S. Comai, and M. Matera. *Designing Data-Intensive Web Applications.* Morgan Kaufmann, 2003.
3. G. Chen and D. Kotz. A survey of context-aware mobile computing research. *In Tech. Rep. TR2000-381, Dartmouth,* 2000.

4. R. De Virgilio and R. Torlone. A General Methodology for Context-Aware Data Access. In *4th Int. ACM Workshop on Data Engineering for Wireless and Mobile Access (MOBIDE'05)*, 2005.
5. Z. Fiala, M. Hinz, K. Meissner, and F. Wehner. A component-based approach for adaptive dynamic web documents. *Journal of Web Engineering, Rinton Press*, (2):058–073, 2003.
6. Z. Fiala, F. Frasincar , M. Hinz, G.J. Houben, P. Barna, K. Meißner. Engineering the presentation layer of adaptable web information systems. In *International Conference on Web Engineering (ICWE'04)*, pages: 459–472, 2004.
7. W. Gu and A. S. Helal. An XML Based Solution to Delivering Adaptive Web Content for Mobile Clients. In *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'04)*, pages: 25–29, 2004.
8. K. Goslar and A. Schill. Modeling Contextual Information Using Active Data Structures. In *EDBT Workshops (PIM'04)*, pages: 325-334, 2004.
9. HP Labs Semantic Web Programme. *Internet document: http://jena.sourceforge.net/*, 2004.
10. J. Indulska, R. Robinsona, A. Rakotonirainy, and K. Henricksen. Experiences in using CC/PP in context-aware systems. In *4th International Conference on Mobile Data Management (MDM'03)*, pages: 247-261, 2003.
11. O. Pastor, J. Fons, and V. Pelechano. A method to develop web applications from web-oriented conceptual models. In *International Workshop on Web Oriented Software Technology (IWWOST'03)*, pages: 144–173, 2003.
12. J. Pascoe. Adding Generic Contextual Capabilities to Wearable Computers. In *2nd International Symposium on Wearable Computers (ISWC 1998)*, pages: 92-99, 1998.
13. D. Schwabe, G. Rossi, and S.D.J. Barbarosa. Systematic hypermedia application design with oohdm. In *The Seventh ACM Conference on Hypertext ACM (Hypertext'96)*, pages: 116–128, 1996.
14. B. N. Schilit, N. Adams, R. Want. Context-Aware Computing applications. In *IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'03)*, 2003.
15. T. Strang, C. Linnhoff-Popien, and F. K. Cool. A Context Ontology Language to enable Contextual Interoperability. In *4th International Conference on Distributed Applications and Interoperable Systems (DAIS2003)*, pages: 236–247, 2003.
16. R. Vdovjak, F. Fransincar, G.J. Houben, and P. Barna. Engineering semantic web information systems in hera. *Journal of Web Engineering, Rinton Press* , (2):003–026, 2003.
17. W3C Working Group on Device Independence. Device Independence Principles. *Internet document: http://www.w3.org/TR/di-princ/*, 2003.
18. W3C. Composite Capabilities / Preferences Profile (CC/PP). *Internet document: http://www.w3.org/Mobile/CCPP/*, 2004.
19. WapForum. User Agent Profile (UAProf). *Internet document: http://www.wapforum.org/*, 2004

Context-Aware Recommendations on the Mobile Web

Hong Joo Lee, Joon Yeon Choi, and Sung Joo Park

Graduate School of Management, Korea Advanced Institute of Science and Technology,
Chengryangridong, Dongdaemungu, Seoul, Republic of Korea
{hjlee, zoon, sjpark}@kgsm.kaist.ac.kr

Abstract. Recently, there has been a significant increase in the use of data via the mobile web. Since the user interfaces for mobile devices are inconvenient for browsing through many pages and searching their contents, many studies have focused on ways to recommend content or menus that users prefer. However, the mobile usage pattern of content or services differs according to context. In this paper, we apply context information—location, time, identity, activity, and device—to recommend services or content on the mobile web. A Korean mobile service provider has implemented context-aware recommendations. The usage logs of this service are analyzed to show the performance of context-aware recommendations.

1 Introduction

The mobile web, which allows users navigate the Web using wireless devices, such as cell phones and personal digital assistants (PDAs), has been gaining in popularity. However, the inconvenient user interfaces of mobile devices may constitute a barrier to browsing through content. Therefore, the mobile web should provide easy access to the services or content preferred by users. Many studies have focused on recommending content that users like, based on user preferences for content acquired from explicit ratings or implicit usage history [2, 8, 11, 12, 15]. However, the mobile usage pattern of content, or services, differs according to factors such as the current time, the user's location, or what they are doing. Few recommendation systems for the mobile web consider context information for personalizing content or services.

In this paper, we apply context information to recommend mobile web services or content. We use location, time, identity, activity, and device information as contextual information. Content ordering, location-based services, cross selling, service filtering, and recommendation pushing are used as recommendation strategies in the mobile web. A Korean mobile service provider has implemented context-aware recommendation services. We analyzed the usage logs of this service to determine the performance of context-aware recommendations and the usefulness of context information.

Section 2 summarizes studies related to context-aware recommendations for the mobile web. Section 3 introduces context-aware recommendations in the mobile web. Section 4 discusses user experiences with context-aware recommendations and analyzes their performance. Section 5 presents the implications and conclusions of this study.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 142 – 151, 2005.
© Springer-Verlag Berlin Heidelberg 2005

2 Context and Recommendations in the Mobile Web

Context is any information that can be used to characterize the situation of an entity, *i.e.*, a person, place, or thing [3]. A system is context-aware if it uses context to provide relevant information or services [3]. Studies on context-aware computing have used location, identity, and time information to make restaurant recommendations [12, 13], to guide tours [2, 11], for advertising [15] or music [5] selection, and to organize schedules [10]. The location is the geographical setting where a person uses the mobile Internet [9]. The time is the current time, according to the system clock of the device used or the service provider [7]. Identity pertains to the current user of the device and his or her preferences. Dey and Abowd added an activity context, which considers what is occurring in a situation. Context-aware applications look at what the user is doing in addition to where, when, and with whom the user is engaging in this activity, and use this information to determine why the situation is occurring and what services or contents the user would favor [3].

Since cellular phones and other mobile devices provide a limited user interface and business logic for clients, there is little use for keyword-based searches; furthermore, browsing between pages is inconvenient [1, 6, 14]. The limitations of cellular phones and mobile devices necessitate personalization in order to recommend appropriate, well-timed content to users. A mobile phone can also provide more direct recommendations via a 'push' service through short message services (SMS) or other interactive channels [6]. Many studies and systems have examined recommendations and personalization on the mobile web using preference-based recommendation methods [8, 15].

Many researchers have defined context in the mobile web and have applied context information to the recommendation of services and content. Hofer *et al.* suggested a context-aware system framework that made use of time, location, users, device, and network-context information [7]. Device distinguishes between different types of device, such as laptops, PDAs, and cellular phones. Network context contains information about the available network connection types the device has. Häkkilä and Mäntyjärvi used the following context information for collaboration in the mobile web: the physical environment, user's goals, device applications, local ad-hoc connections, and connections to infrastructures [4]. Lee *et al.* classified use context for the mobile Internet into personal and environmental context: personal context included time, movement, and the user's emotion; environmental context included the physical context of users, such as location, crowding, and distraction, and social context, such as interaction and privacy [9]. Yuan and Tsao used time, place (outdoors or indoors), fee (free or fee-based), and other content-based information to contextualize mobile advertising [15]. Presently, mobile service providers operate various location-based services, such as 'searching for friends', which notifies users about where their friends are, or to make recommendations about nearby restaurants. Context-aware recommendation systems and studies are summarized in Table 1.

Table 1. Context-aware recommendation systems and studies

Literature and year	Type of Recommendation	Context
Cheverst *et al*. 2000 [2]	Tour information	location, user interest, time, operating hours of attractions
Tewari *et al*. 2002 [12]	Restaurant	location
Brunato *et al*. 2003 [1]	Location-aware content	location, navigation history
Yuan *et al*. 2003 [15]	Advertising	location, time, fee, content attributes
Pousman *et al*. 2004 [10]	Schedule	location
Hayes *et al*. 2004 [5]	Music	user interest, situation
Tung *et al*. 2004 [13]	Restaurant	location, time, weather
Setten *et al*. 2004 [11]	Tour information	location, time, weather, agenda, shopping list

3 Context-Aware Recommendations on the Mobile Web

3.1 Context of the Mobile Web

Using the existing classification and definition of mobile context, as well as related research, a hierarchy of mobile context was developed, as shown in Fig. 1. At the top level, contextual information is divided into five categories: location, identity, activity, time, and device. Device context is composed of the features of mobile devices and the network connection to the mobile Internet. Important features of mobile devices are display (color or gray), audio (polyphonic ringer, mp3), and video (camera, camcorder, and digital multimedia broadcasting (DMB)) capabilities, and storage capacity. Network connection to the mobile Internet concerns the type of cellular technology and wireless interface, such as GSM, CDMA, and Bluetooth.

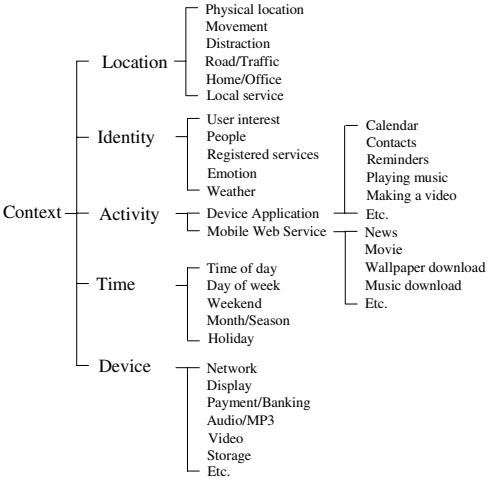


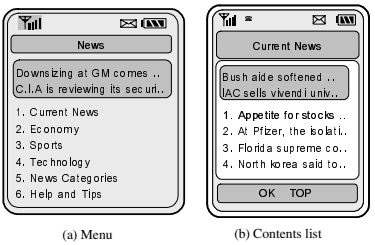
Fig. 1. Context hierarchy of the Mobile Web

Location represents not only the current position of the user, but also who is near the user, what objects are near the user, such as distractions, roads, and local services, and whether the user is moving or staying at a specific place, such as the home or the office. Identity represents personal interests about services or content and people, *e.g.*, usage history, registered services, preferred keywords, emotional state, and social network. Time is the current time, as given by the system clock of the device used or server operating; other factors can be determined from the current time, such as day of week, weekend, season, and holiday. Activity describes the current use of the mobile device and mobile web, such as what applications of the device the user is exploiting and what kind of mobile service or content the user is using.

3.2 Context-Aware Recommendations for the Mobile Web

In this section, we summarize context-aware recommendations that can be applied to the mobile web. The following are recommendation strategies used on the mobile web: content ordering, location-based services, cross selling, service filtering, and recommendation pushing.

Content Ordering. Content ordering includes menu ordering and content list ordering, as shown in Fig. 2. Mobile web pages are organized hierarchically, *e.g.*, top menu, submenus, contents list, and content viewing page. Therefore, context-aware recommendations can be used to show personalized menu structures and a content list may be set up, according to the user's preferences and context. For example, a user may see economy and business news categories in the morning, while the user tends to see entertainment news after lunch.



(a) Menu (b) Contents list

Fig. 2. Examples of screens on the mobile web

Location-Based Services. Table 2 is a set of location-based services operated by Korean mobile service providers. Some location-based services are not suitable for applying context-aware recommendations, but recommendations can be applied to the 'Best Restaurant' and 'About Town' services based on the user's identity and other contextual factors. After the current location of the user is detected, then the recommendation problem becomes that of providing lists of objects near the user.

Cross Selling. As Fig. 2 shows, a promotion or recommendation can be located in the upper part of the screen. This area is used for notifying the user about events and promotions, and for recommending 'hot' content in the menu. Personalized cross

Table 2. Location-based services provided by Korean mobile service providers

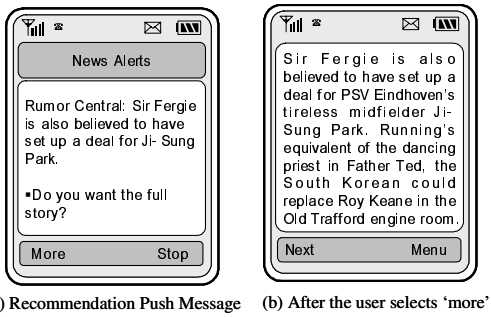
Service Title*	Description
Search for Friends	Notifies users where their friends are (if the users and their friends are registered with this service)
Best Restaurant	Makes recommendations about nearby restaurants
Location Notification	Tells parents or police the users' location (an emergency service for children)
About Town	Advertises events or special promotions by off-line shops in the users' favorite areas
Personal GPS	Shows the shortest paths to the destination by walking or public transportation
Traffic Navigation	Shows current traffic information and the shortest path to the destination by driving

* The authors have translated some of the service names from Korean into English.

selling is done by selecting relevant items belonging to other categories or menus, based on content in the current menu. Recommended cross-selling items can be placed in this region to stimulate the usage of other mobile web content. In addition, relevant items can be placed below the currently used content. For example, when a user reads an article on a newly released movie, presenting information on movie ticket reservations or movie services may help the user.

Service Filtering. Not all users or devices can use all services or content, *e.g.*, non-mp3 devices cannot launch music-streaming or mp3-downloading services. Therefore, the capabilities of the user's device and services registered to the user should be considered when making recommendations on the mobile web.

Recommendation Pushing. Existing web pages and mobile web pages are pull-based services that are invoked when users enter the pages. However, the mobile web has a direct recommendation channel that uses a 'push' service through SMS and multimedia messaging service (MMS). This service can be an active channel for delivering personalized content to the user. Before mobile service providers send personalized push messages, they have to decide what services or content should be delivered and what context is suitable for sending a specific push message, such as the



(a) Recommendation Push Message (b) After the user selects 'more'

Fig. 3. Screen examples of recommendation push

time of the day, day of week, location of the user, and other factors. Since the usage patterns of users differ according to context and content, there is a preferred context for receiving and responding to push messages.

Figure 3(a) is an example of a recommendation push delivered to a user and Fig. 3(b) shows an example of the screen that appears when the user selects 'more'. Receiving recommendation messages are free of charge; after push 'more' button, usage fee is charged based on received packets.

3.3 System Framework

The overall framework of the context-aware recommendation system is shown in Fig. 4.

Five main groups can be identified in the framework: the *interaction manager*, *service provider*, *recommendation engine*, *context manager*, and *profilers*. The *interaction manager* catches users' requests, invokes services, publishes mobile web pages, and delivers recommendation messages. The *service provider* has two parts: the *service manager* publishes mobile web pages and displays content on pages, as well as recommending content generated by the *recommendation engine*. The *messaging manager* transfers recommendation push messages generated by the recommendation engine to a messaging infrastructure and monitors the context of the user for sending recommendation messages.

Usage logs of users, content, and user databases are fundamental sources for building *user profiles* and *service profiles*. A *user profile* describes user preferences for services and context, characteristics of the user, and user segments, which are small groups of similar users. The *service profile* describes the usage pattern of the users according to context, *e.g.*, when each service is most used, and where each service is most used, such as at home, the office, or downtown.

The *context manager* retrieves information about the user's context by contacting the appropriate context services, such as the location monitor, which records the current location of the user, time, device, and activity, periodically. It is also

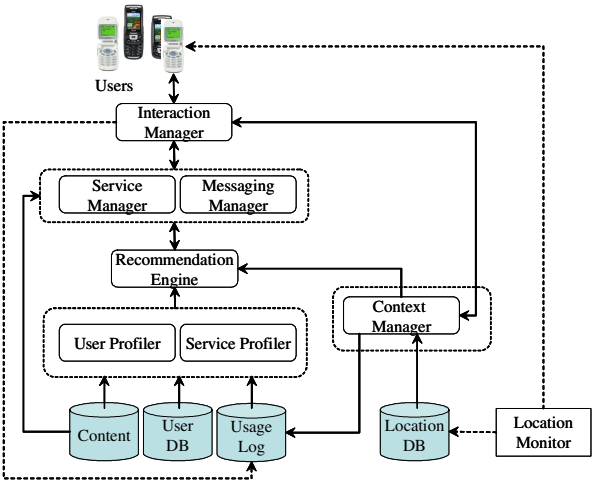


Fig. 4. The system framework for context-aware recommendations

responsible for reasoning the user's context information. For example, the location of the user's home is derived by aggregating location information from midnight to dawn for 30 days, and the movement of the user can be identified by tracing location records of the user in the location database.

The *recommendation engine* uses multiple prediction strategies to predict how interesting each mobile web service and content is to the user, based on user profiles and service profiles. Recommendation strategies select suitable content to show the user at the top of the menu screen or at the bottom of the contents screen, and rank submenus and the contents list. Used recommendation strategies include collaborative filtering, rule-guided filtering, association rules, and sequential pattern. The most important input sources for the recommendation engine are the *user profile* and *service profile* generated by the profilers. Recommendation history and recommendation rules are applied to the *recommendation engine*, *e.g.*, the same category content should not be recommended within a certain period.

4 User Experiences

The use of context-aware recommendations was analyzed for 200 users registered with a Korean mobile service provider's intelligent wireless service. Usage logs of news, movie, and restaurant services from March 2005 until the end of May 2005 were analyzed to show the performance of context-aware recommendations. Fig. 5 shows the usage patterns by time of day and day of week. As you can see there are fluctuations of usage through time of day and day of week. Also, there are usage peaks around 10 a.m. and 3 p.m. each day, and the usage percentage is similar on weekdays. On weekends, the users did not use mobile content or services as much as on weekdays.

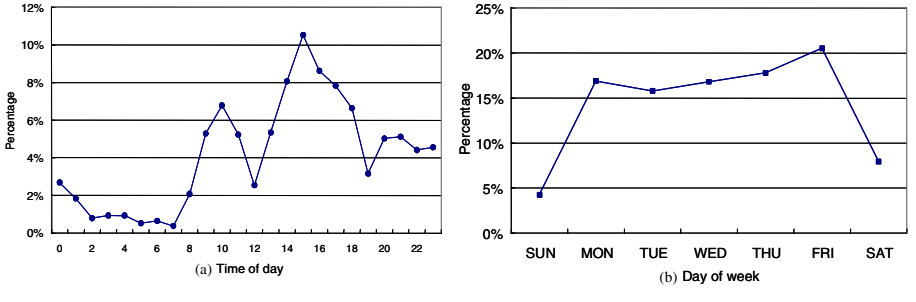


Fig. 5. Usage percentage for the mobile web by time of day and day of week

Of the many recommendation strategies used on the mobile web, we analyzed the performance of recommendation pushing. A screen example of a recommendation push message is shown in Fig. 3. If the user selects the 'more' button after receiving a push message, it is recorded as positive feedback. If the user selects the 'stop' button, it is recorded as negative feedback. We used the following equation as a measure of precision:

$$Precision = (Number\ of\ instances\ of\ Positive\ Feedback)/(Total\ Number\ of\ Push\ Messages) \quad (1)$$

First, we built user profiles and service profiles from the usage log for March and April 2005. User profiles represent each user's service preference for context, and service profiles show all the users' service preferences for context. The service level can be varied to adapt to the recommending level of a service, such as menu, content category, or content. Then, we select preferred context to send a push message to a certain user using the integrated weighted sum of the user and service profiles. The weights for the user and service profiles were calculated from the entropy of the profiles. If the user profile shows a solid preference for a certain context variable, the entropy value of the user profile is increased. The messaging manager monitors the current context of the user and sends the push message when the user is in the selected context. Table 3 shows the average precision of the push messages that were delivered to users in the preferred context and not preferred context.

Table 3. Average precision by context variable

Context		Precision		t-value
		Average	S.D.	
Time of Day (24 hours)	Preferred daily time	0.579	0.399	6.762 (Sig. = 0.000)
	Not Preferred daily time	0.145	0.325	
Day of Week (7 days)	Preferred weekly time	0.591	0.373	9.773 (Sig. = 0.000)
	Not preferred weekly time	0.079	0.243	
Location (zip codes)	Preferred location	0.637	0.338	5.600 (Sig. = 0.000)
	Not preferred location	0.377	0.308	

* $\alpha = 0.05$

When the user location and selected location for the push message are the same, the average precision of the push messages was highest. The average precision of the other contexts was also over 0.5. This means that more than half of the users who received recommendation push messages selected the 'more' button to see the recommended content or use the recommended service. We used paired t-tests to compare differences in responses in the two conditions (preferred vs. not preferred context). For each of the three types of context, the precision of push messages sent in the preferred context was higher than that of messages sent in a not preferred context. This difference was greatest for day of week.

5 Conclusions

This paper presented a context-aware recommendation system for the mobile web, a system that incorporates context awareness, user preference, and service preference to provide recommendations. We defined location, time, identity, activity, and device information as contextual information in relation to the mobile web. Content ordering, location-based services, cross selling, service filtering, and recommendation pushing are suggested as recommendation strategies in the mobile web. The overall framework of the system has been introduced. The major input sources for the

recommendation engine are the user and service profiles for context, and the context manager, which retrieves information about the user's context by contacting the appropriate context services. A Korean mobile service provider has implemented context-aware recommendation services. The usage pattern of mobile web services was analyzed and the performance of context-aware recommendations, and the usefulness of the context information were analyzed. The average precision of push messages, which are delivered when the context of the user is the same as the selected context, was over 0.5.

Context-awareness and recommender systems can enhance and complement each other in that they help users to find relevant and interesting items, according to their interests and context information [11]. Although recommendation and context-awareness systems are useful tools that may be used to increase customer loyalty and response rates, mobile service providers should not make too many recommendations, or they will irritate users.

References

1. Brunato, M., Battiti, R.: PILGRIM: A Location Broker and Mobility-Aware Recommendation System. In: Proceedings of IEEE PerCom2003, Dallas-Fort Worth, Texas (2003) 1-8
2. Cheverst, K., Davies, N., Mitchell, K., Friday, A.: Experiences of Developing and Deploying a Context-Aware Tourist Guide: The GUIDE Project. In: MOBICOM 2000, Boston, MA, USA (2000) 20-31
3. Dey, A.K., Abowd, G.D.: Towards a Better Understanding of Context and Context-Awareness. In: CHI 2000 Workshop on The What, Who, Where, When, Why and How of Context-awareness, the Hague, the Netherlands (2000) 1-6
4. Häkkilä, J., Mäntyjärvi, J.: Collaboration in Context-Aware Mobile Phone Applications. In: the 38th Hawaii International Conference on System Sciences, Hawaii, USA (2005) 1-7
5. Hayes, C., Cunningham, R.: Context Boosting Collaborative Recommendations. In: Knowledge-based Systems, 17 (2004) 131-138
6. Ho, S.Y., Kwok, S.H.: The Attraction of Personalized Service for Users in Mobile Commerce: An Empirical Study. In: ACM SIGecom Exchanges, 3, 4 (2003) 10-18
7. Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, G., Altmann, J.: Context-Awareness on Mobile Devices - the Hydrogen Approach. In: the 36th Hawaii International Conference on System Sciences, Hawaii, USA (2003) 292
8. Kim, C.Y., Lee, J.K., Cho, Y.H., Kim, D.H.: Viscors: A Visual-Content Recommender for the Mobile Web. In: IEEE Intelligent Systems, 19, 6 (2004) 32-39
9. Lee, I., Kim, J., Kim, J.: Use Contexts for the Mobile Internet: A Longitudinal Study Monitoring Actual Use of Mobile Internet Services. In: International Journal of Human-Computer Interaction, 18, 3 (2005) 269-292
10. Pousman, Z., Iachello, G., Fithian, R., Moghazy, J., Stasko, J.: Design Iteration for a Location-aware Event Planner. In: Personal and Ubiquitous Computing, 8 (2004) 117-125
11. Setten, M.C., Pokraev, S., Koolwaaij, J.: Context-Aware Recommendations in the Mobile Tourist Application COMPASS. In Nejdl, W. and P.D. Bra, Editors (eds) AH 2004, LNCS 3137, 2004. p. 235-244
12. Tewari, G., Youll, J., Maes, P.: Personalized Location-based Brokering using an Agent-based Intermediary Architecture. In: Decision Support Systems, 34 (2002) 127-137

13. Tung, H.-W., Soo, V.-W.: A Personalized Restaurant Recommender Agent for Mobile e-Service. In: IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04), Taipei, Taiwan (2004)
14. Varshney, U.: Location Management for Mobile Commerce Applications in Wireless Internet Environment. In: ACM Transactions on Internet Technology, 3, 3 (2003) 236-255
15. Yuan, S.-T., Tsao, Y.W.: A Recommendation Mechanism for Contextualized Mobile Advertising. In: Expert Systems with Applications, 24 (2003) 399-414

Capturing Context in Collaborative Profiles

Doris Jung and Annika Hinze

University of Waikato, Hamilton, New Zealand
{d.jung, a.hinze}@cs.waikato.ac.nz

Abstract. In various application areas for alerting systems, the context and knowledge of several parties affect profile definition and filtering. For example, in healthcare nurses, doctors and patient influence the treatment process. Thus, profiles for alerting systems have to be generated by the explicit collaboration of several parties who may not know each other directly.

We propose the new concept of *collaborative profiles* to capture these different conditions and contexts. These profiles exploit each single party's expert-knowledge for defining the context under which (health-related) alerting is required. Challenges include the definition and refinement of profiles as well as conflict detection in context definitions.

1 Introduction

Users of an alerting system are notified about events or data that are of interest to them. A user's interests are captured in a profile; incoming information is filtered according to the user's profile (see Fig. 1, top). An emerging application area for alerting is the healthcare sector, where an alerting system could notify about critical results of heart rate measurements or a blood value sensor. In [4], we suggest a mobile alerting system for supporting patients in the management of their chronic conditions. In the design of such a patient-centred alerting system, we encounter several challenges [5]. Some of these challenges can also be found in other application areas, e.g. tourism, e-commerce and facility management.

In this paper, we focus on two challenges in particular: Different to typical alerting systems, (i) profile definitions should exploit the expert-knowledge of several parties participating in patient treatment, and (ii) the functionality of alerting systems has to adapt to different contexts to allow for heterogeneity in patients and healthcare providers. Addressing both challenges, we propose to exploit the concept of *context-awareness* in profile definition and evaluation. Patients and clinical staff create profiles collaboratively but consecutively (context is defined by *several* parties). The actions specified in these profiles adhere to each patient's personal and healthcare background (context influences system functionality). We introduce the concept of *collaborative profiles* and the principles of the collaborative exploitation of a user's expert-knowledge.

The paper is structured as follows: We first describe our concepts used for collaborative profiles followed by an initial classification of collaborative profiles. We briefly compare our approach to related work and conclude the paper by outlining future plans.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 152–155, 2005.
© Springer-Verlag Berlin Heidelberg 2005

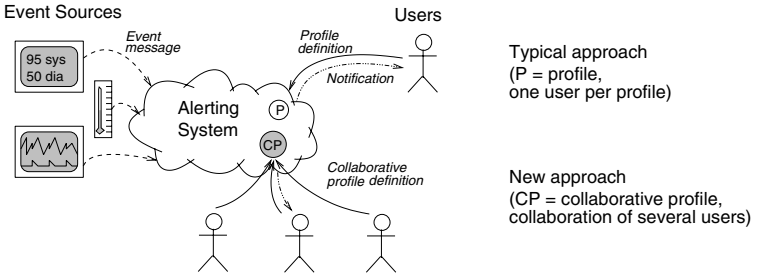


Fig. 1. An alerting system extended with collaborative profiles

2 Concepts

In this section, we explain our concept of collaborative profiles and the notions required for its understanding.

Context: We base our definition of context on Dey, i.e. context is any information that can be used to characterise the situation of an entity [3].

Collaboration: Similar to the area of CSCW, our understanding of collaboration is that of work which is undertaken jointly with others. This may include collaborating partners which are not immediately connected.

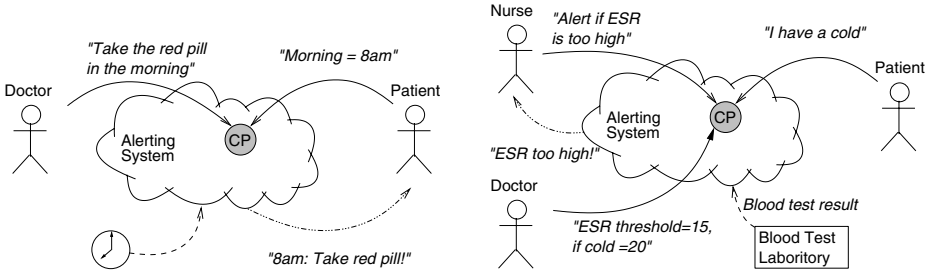
Collaborative Profiles: As identified in [4,5], in several application domains profiles have to take into account the expert-knowledge and the specific contexts of several users. We therefore propose that several parties collaboratively define profiles. Profiles also determine the filter criteria for the alerting system. Thus, collaborative profile definition leads to a collaborative specification of the filter functionality to ensure a context-based profile evaluation.

Fig. 1 (bottom) shows the idea of several users collaborating for the profile definition: Several users share their knowledge and collaboratively specify and refine the profile. The users to be notified are selected according to context. For example, after an initial definition of a patient's profile, other doctors and nurses subsequently refine the profile when the patient visits them. Depending on the patient's current condition (i.e. changing context), the patient may be alerted to take specific medicine, or doctor and patient may be alerted to a serious change in the patient's condition.

Collaborative profiles may have to undergo several refinement steps as they may contain uncertain or vague specifications (e.g. regarding time or health condition) which have to be refined by other health practitioners as the patient visits several specialists in the course of time.

3 Classification of Collaborative Profiles

We now introduce our initial classification for types of collaborative profiles, which lays the foundation for the system's implementation design.



(a) Refinement of vague profiles (b) Merging of N specifications ($N > 2$)

Fig. 2. Types of collaborative profiles (1. Refinement – left; 2. Merging – right)

1. **Refinement:** A user defines a profile that contains parameters, which are not precisely defined but refer to semantically vague concepts. Another user, who has special knowledge the first user lacks, then provides the missing information and thereby refines the profile. An example is presented in Fig. 2(a): The doctor specifies that the patient should take a red pill in the morning (general context). However, only the patient knows exactly what time she considers to be morning (specific context), so she refines the time as 8am. At 8am the patient is alerted to take the medicine.
2. **Merging:** If more than two users take part in the process of profile definition, two cases can be differentiated: Either the specifications provided by the users can be merged to a single profile, or two or more of the specifications are conflicting. For the latter case refer to Type 3. The possibility of merging the specifications is illustrated in Fig. 2(b). Three users are shown: a nurse, a doctor and a patient. The nurse defines a profile for being notified whenever the patient's ESR (an inflammation marker) is too high. The doctor has more detailed knowledge and defines the actual ESR threshold as 15, or, if the patient has a cold, as 20. The patient knows best about his current health condition and indicates whenever he as a cold. The profile is evaluated depending on the patient's context, and the nurse is notified if necessary.
3. **Conflict:** The profile specifications given by two or more users may be contradicting and lead to conflict. This can happen in situations such as the following: One doctor recommends alkaline food for the patient due to condition A (context A), whereas another doctor recommends acid food due to condition B (context B). It is important that such situations are detected: The users have to be alerted of contradicting context definitions so that they can refine or re-define the profiles.
4. **Personalisation:** For short-term context changes, alerting systems may provide default profiles that can be personalised (here, we give an example from facility management). Such profiles could be provided by building managers and can control the temperature of a heating or the brightness of a room. Each tenant can then fine-tune and personalise their profile, e.g. tenants adjust their room temperature. This is especially applicable for short-term

context changes (e.g. special requirements for art exhibitions). Once the short term context ends, the parameters could be reset to their original values.

4 Related Work

Our initial analysis of related work has revealed that collaborative profiles are a novel concept to alerting systems. So far, it has neither been exploited for integrating the user's expert-knowledge nor for capturing various and changing user contexts. Systems for collaborative work, e.g. [2], do not support collaborative profiles. [1] targets collaborative query techniques and is not applicable for alerting. Inspiration might be found in a model for uncertainty in alerting systems [6]; nevertheless, probabilistic concepts cannot fully satisfy the requirements of collaborative profiles.

5 Conclusion and Future Work

We have identified the need for the users of alerting systems to collaboratively contribute to profile definitions. These collaborative profiles can capture the rich expert-knowledge and various contexts that are prevalent in healthcare environments. We have illustrated how a mobile alerting system may be used by different parties, such as doctors, nurses, healthcare staff and patients whose context may change temporarily. We have introduced the concept of collaborative profiles and have described our initial classification of types of collaborative profiles.

Currently, we are interviewing healthcare providers to learn more about their contextual background and to analyse their requirements regarding collaborative profiles. The next step is the implementation design of collaborative profiles for our mobile alerting system for patients with chronic conditions based on the classification introduced here.

References

1. A. F. Blackwell, M. Stringer, E. F. Toye, and J. A. Rode. Tangible interface for collaborative information retrieval. In *Proc. of CHI 2004*, Apr. 2004.
2. B. G. Buchanan, G. Careini, V. O. Mittal, and J. D. Moore. Designing computer-based frameworks that facilitate doctor-patient collaboration. *Artificial Intelligence in Medicine*, 12(2):169–191, 1998.
3. A. K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001.
4. D. Jung and A. Hinze. Alerting system for the support of patients with chronic conditions. In *Proc. of the European Conference on Mobile Government (EURO mGOV)*, Brighton, UK, June 2005.
5. D. Jung and A. Hinze. Patient-based mobile alerting systems - requirements and expectations. In *Proc. of the Health Informatics in New Zealand Conference (HINZ 2005)*, Auckland, NZ, Aug. 2005.
6. H. Liu and H.-A. Jacobson. Modeling uncertainties in publish/subscribe systems. In *Proc. of the International Conference on Data Engineering (ICDE'04)*, Aug. 2004.

Using Context of a Mobile User to Prefetch Relevant Information

Holger Kirchner

Fraunhofer IPSI, Dolivostrasse 15, 64293 Darmstadt, Germany
holger.kirchner@ipsi.fraunhofer.de

Abstract. Providing mobile users with relevant and up-to-date information on the move through wireless communication needs to take the current context of a user into account. In this paper, the context of a user with respect to his movement behaviour as well as device characteristics is under investigation. In outdoor areas, particularly in an urban area, obviously there is often sufficient communication bandwidth available. In some areas though, especially in rural areas, communication bandwidth coverage is often poor. Providing users in such areas with relevant information and making this information available in time is a major challenge. Prefetching tries to overcome these problems by using predefined user context settings. In situations where resource restrictions like limited bandwidth or insufficient memory apply, strategies come into place to optimize the process. Such strategies will be discussed. Evaluating the different types of users supports the approach of getting the relevant information to the user at the right time and the right place.

1. Introduction

1.1 Motivation

As people become more nomadic, there is an increased need for constant access to information while on the move [1]. Mobile devices supporting wireless communication and mobile networks are becoming ubiquitous. The networks deployed today (e.g. GSM, GPRS) are generally providing data transmission of very limited and low bandwidth. Major additional drawbacks of wireless communication networks are high delays [2], frequent disconnections [3] and users that do not always want to be connected.

New devices are already equipped with different network adapters such as IEEE 802.11b (WiFi), Bluetooth [4], GPRS or UMTS [5], which allow mobile users simultaneous connectivity, thereby ensuring higher network access availability. People also often want to retrieve information and use services in their vicinity with respect to their location, time and personal preferences [1]. Information use often depends on the properties of a mobile device itself, e.g. screen size or limitations of the main memory. A mobile device may disconnect voluntarily (to save power or connection cost) [2] or involuntarily (by failure or coverage disturbance) from the network. At the same time users require seamless communication and data access [6] thus guaranteeing a good user experience, i.e. fast and reliable information access at all times. This a major challenge. Prefetching is an elegant technique for handling information

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 156 – 165, 2005.
© Springer-Verlag Berlin Heidelberg 2005

provision for users under these circumstances. This idea though is not new and has already been proposed e.g. in [6]. Prefetching has mostly been done on unstructured data and has focused on access probabilities on files. Location as one parameter of a context has been investigated in a few, such as [8] and [9]. As a consequence context-awareness turns out to be very important for delivering relevant data from different sources.

In this paper we describe an approach for prefetching that supports mobile users in a highly dynamic environment. Our approach supports the use of the current location, the time as well as the movement pattern and the user profile data (user's interests, user's preferences) to provide more relevant information to the user.

1.2 Sample Application Area

The concept of context-aware prefetching is applied to a tourist information system for boaters so called eureauweb[1], where boaters require information on the move. This application area provides exactly the characteristics mentioned above. The information within the system can be represented by 'objects' with a set of attributes like geographical position and some categorization. To measure the relevance of the information, we use the geographical position and some predefined categories as measurements as well as access probabilities for particular categories.

1.3 Outline

In the next section, related work is reviewed. Section 3 explains the fundamentals of prefetching. Section 4 defines the context and in section 5 the prefetching process is explained. In section 6 some experiments are given, before some conclusions are drawn in section 7.

2 Related Work: Hoarding and Info Stations

Projects described in [3], [7] and [8] provide users with information over wireless links. They are generally based on info stations or hot-spotted areas. The idea is to provide the mobile clients with data at specific locations that provide access to WLAN or other high-bandwidth infrastructures. The prefetching architecture introduced by [7] provides postfetching from a client's point of view. Users can demand voluminous data while between two hot-spots. A prefetch agent gathers the data in the name of the user. As soon as he later enters the scope of a high-bandwidth area, the data is forwarded to the user. Hence, prefetching is fulfilled between the services and the prefetch agent and not between the services and the mobile user.

[3] and [8] apply data hoarding, i.e. they store data on the mobile devices before leaving the info-station. The difficulty lies in the prediction of the data needed on the way from one info station to another, i.e. on the way to the next high-bandwidth network access point. [3] offers mapping information for car drivers in California. The

[1] The eureauweb™ project, which is partly funded by the EU under the IST-2001-I.5.3 action line for ambient intelligence application systems for mobile users and travel/tourism businesses, aims to develop an information system for European Inland Waterways.

mobile devices are provided simultaneously with more or less detailed maps for a predefined itinerary depending on the user's location, direction and speed. [9] offers an architecture for mobile tourist guides. They use info-station visiting maps and external static maps with absolute and relative probabilities to compute the most probable user paths. A visit probability map is calculated together with the access probabilities of the individual data items. The items with the highest probability are then transferred to the mobile unit. The problems of these algorithms are obviously directly related to the hoarding problem. Data that is not present on the mobile devices when leaving the range of an info-station can either not be accessed at all or has to be downloaded through high-latency WANs from scratch. Comparing these existing approaches to our requirements, we see two major differences:

Independent of any infrastructure – the physical location and the range of the signal defines the area where logical information is valid.

Support of low bandwidth communication – info-stations or hot spots require high bandwidth communication links.

3 Fundamentals of Prefetching

The principle of reducing response times can be simply described by figure 1.

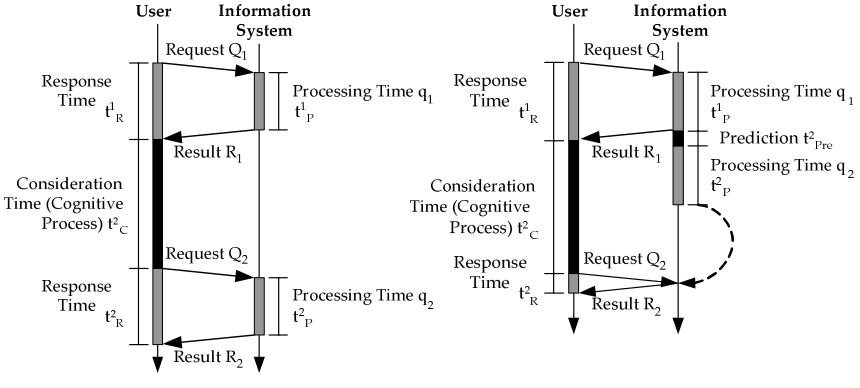


Fig. 1. a.) Non-prefetching solution and b.) Prefetching solution

Users have a need for information. They send a query Q_1 to the information system and after some processing time t^1_P, the result R_1 will be returned to the user. After the consideration time t^2_C, which is described here as a cognitive process, the user might send a new query Q_2 to the system. The difference to a prefetching solution is shown in figure 1b. During the consideration time the system is automatically pre-processing a query Q_2 which it considers to be relevant to a user. When a user queries such pre-processed information the response time is much reduced by t^2_R.

While the query of the system is being transmitted to the information system, executed and the results are sent back to the user; the user waits (this is the latency time perceived by the user). Afterwards, a cognitive process takes place when the user

evaluates the result of the last query and builds a new suggestion. For our illustration, we assume the system is idle during the *consideration time* (after which the user executes Q_2) If Q_2 can be predicted using information about the context then the query can be executed during the cognitive process. If the prediction was correct, the next query can directly be answered from the cache, reducing the response time perceived by the user. If the user executes the query while the execution of the query is still in process (that means $t^2_{Pre} < t^2_C < t^2_{Pre} + t^2_P$), the user's latency time is still shorter compared to the non-prefetching solution by $t^2_C - t^2_{Pre}$ as the query is already running when the user executes it. If the consideration time is even shorter than the prediction time $t^2_C < t^2_{Pre}$ no speed up is achieved.

This basic concept can be taken for stationary as well as for mobile environments. The major difference is obviously that processing time for t^1_P and t^2_P is dominated by communication delays.

4 Defining the Context

Within the context of mobile information systems, the specific information is represented as objects and, in our application area, as a point-of-interest (PoI). As described in [1], there are a set of categories ranging from navigational information to information about boat services, accommodation, eating & drinking, shopping, tourism, entertainment, infrastructure and practical use. All of these types of information are relevant for boaters and needs to be addressed from the mobile side. In our approach we use the River Coordinate System (RCS) [10], which maps the PoI to location models by finding the closest distance to a waterway.

Several location models have been studied in order to evaluate target zones *tz*. A target zone describes a particular area of geographic relevance of objects for a user which changes over time.

A raw position gives the current latitude/longitude (see the points in figure 2). The target zone is used to define an area of validity for objects which is usually done by a spatial query. In our approach, we investigate not only the location of a user; we also take direction and velocity to find an appropriate area of information validity.

For info-stations as defined in [8] a cell itself describes a target zone, which can be uniquely identified by cell-IDs. There are in general two problems with this approach. First, accuracy and second relationships between cells. Accuracy varies between a few hundred meters and several kilometers. So the information distribution is based on a size of a cell. In places where there is no base station, no mapping of relevant information can be made (see 2a). The direction of a user can theoretically be handled by locating neighbourhood cells for a particular direction. In practice this is not useable because there is no global model available which supports an addressing scheme over network operators. Velocity can only be handled by looking more cells forward and direction is pointing to relevant cells (see 2b). Other solutions are geometric shapes, which have been investigated as well. The major advantage is that the target zone can smoothly be controlled by radius r of circles in relation to velocity of a user (see 2c).

The problem is only that if the speed increases the radius, there is a lot of extra information transferred to the user, which is probably not needed. That is why we

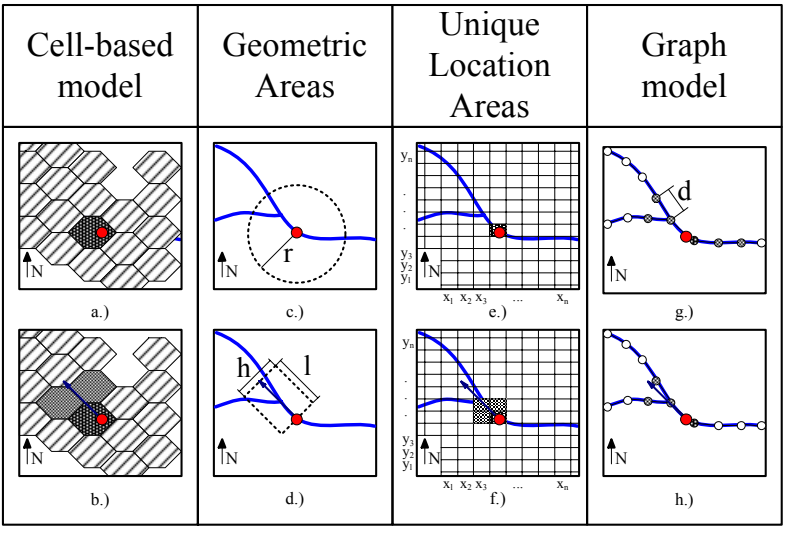


Fig. 2. Modeling locations and defining Target Areas

proposed in [10] an approach which is based on a rectangular shape (see 2d). The benefit is that if the user moves faster the information which is nearby the waterways will not be transferred to the client, because *h* is getting smaller. Instead, we are looking forward more and increasing *l*. A more sophisticated solution is provided by location area codes (LAC) [11]. LACs are unique location areas, which are transformed by latitude/longitude into values between 0 and 1. These values are mapped into a 16 Bit coding scheme and stored as hexadecimal representation. This has a major advantage, because all objects are fixed mapped to an area and a query by an LAC can directly access it (see a). Directions can easily be supported, because relationships between neighbours are given in the schema as well (see 2f).

Another option is a graph model which defines rivers as polylines for our application (see 2g-h). Rivers are *LocationNodes* connected via *LocationEdges*.

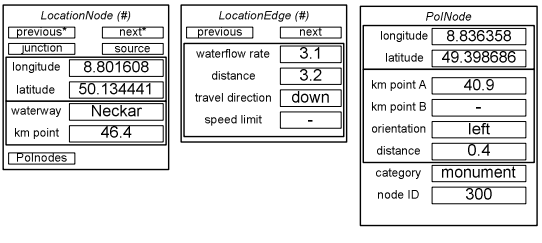


Fig. 3. Model of Location Node, Edge and PoI. In more detail, each *LocationNode* might have multiple *LocationEdges* before (previous*) and after (next*). Additionally, direct access via pointers to the *junction* and the *source* is included as well as the geographical position, the waterway's name, the km point (a pointer on a vector which goes from 0 to max) and a pointer *PoInodes* for attaching PoIs to the *LocationNode*. *LocationEdges* describe the waterflow rate of a particular edge of a river, the distance and the allowed travel direction.

5 Prefetching Process

Prefetching is a possible technology to overcome the above mentioned problems of communication, limited main memory and context-dependent information needs. The idea is to transfer information, which may be needed by the user in the future, in advance, so that it is already stored on the user's mobile device when it will actually be accessed. This has two major advantages. First, the system has *lower response times* because it produces *more cache hits*. Second, *information requisitioning* which *is highly volatile* on the network can be reduced because data is only transferred when enough bandwidth is available, rather than on demand. The system is thus able to provide seamless communication and information access, while creating the illusion of a link with considerably higher bandwidth than is actually the case. On the other hand, there are three main costs involved with prefetching: CPU use, use of main memory, and higher network traffic.

The first involves the CPU cycles expended by the prefetching mechanism in determining which data to prefetch at what moment. The difficulty at this point is to have enough insights available in order to improve the probability that e.g. a user is reaching a certain location or has a need for a particular piece of information. Second, caching the data requires space in the main memory. Prefetching must ensure that data invalidation strategies are applied to get rid of unneeded information. Such a strategy must ensure that there is also memory left for queries which do not fit into a prefetched area. Cycles are spent both on overheads in gathering the information that is necessary to make prefetch decisions, and on actually carrying out the prefetch. The third expense is the network bandwidth and server capacity wasted when prefetch decisions inevitably proven less than perfect.

To avoid excessive network traffic and prefetching cycles, the mechanism has to consider different strategies to increase the efficiency of the algorithm and the relevance of the fetched data. A powerful tool to address the prefetching problem in the context of mobile clients is location-awareness. By using the current user location, direction and possible movement patterns it is possible to significantly restrict the area relevant to a user at a given moment. Due to a smaller amount of relevant data to fetch it is possible to decrease the bandwidth and CPU power required for beneficial prefetching.

In an information system, the information is usually represented by some 'objects' with a finite set of arbitrary attributes. Therefore, the data provided to information system users is defined by a multi-dimensional parameter space. Two dimensions define the geographical area of interest to the user and its current location, while the others represent the content describing attributes. In a way, prefetching is filtering all the possible information within the multi-dimensional information space and then transferring the remaining data. Knowledge about a user's habits, preferences and interests is indispensable to compute a content's importance to a user (and can also be a specialization of a strategy). The mobile user needs to influence the query process explicitly whenever possible. This influence can be of a direct or a more indirect form. In the first case, the user would for example explicitly tell the system that information belonging to a category is of high importance to him. In the latter case, the system would analyze the user's habits, for example by applying some movement

patterns to the user's movements and by deducing the importance of the content from the movement patterns.

Prefetching data for information management demands garbage collection. The best prefetching strategy is of no use without intelligent data caching on the mobile device. Due to the limited storage space available on most of today's mobile devices, only relatively small cache memories are being offered. Thus, a location and data-aware cache invalidation scheme is defined to support the prefetching mechanism. In addition to the invalidation scheme an efficient storage structure and content investigation method must be provided. Only with a cache implementation well-suited for the mobile environment can the perception of wireless links with high bandwidth be created.

The prefetching process consists of a prediction process, cost/benefit estimation and a decision process (figure 4 gives an overview).

The *prediction process* takes the current location, direction and velocity of a user and calculates the target zone (as described in the previous section). The time factor is used to calculate more critical information and to set the priority. User interests reduce the number of categories and device characteristics such as limited main memory should be taken into account. (In this paper we only concentrate on main memory but screen size and other parameters are possible extensions).

Cost / benefit estimations are used to influence the decision process. Costs are given by the rate of data which needs to be transferred by giving the target zone and selection of categories from the source to the client. For updates, we transfer object IDs and modified value in order to reduce traffic of data, which is already accessible locally. It is important to measure the cost for main memory, which appears when fast response times should be achieved. An evaluation of a successful search query (hit) is difficult to measure, because this requires some assumptions. We assume that users are mostly searching for information in their vicinity. In our system, we also rate queries as a successful hit, which can be answered by preloaded data.

Before we can do any experiments a prefetching policy needs to be defined. The *decision process* decides when information is transferred, which information to prefetch and if any data in the cache needs to be replaced or not.

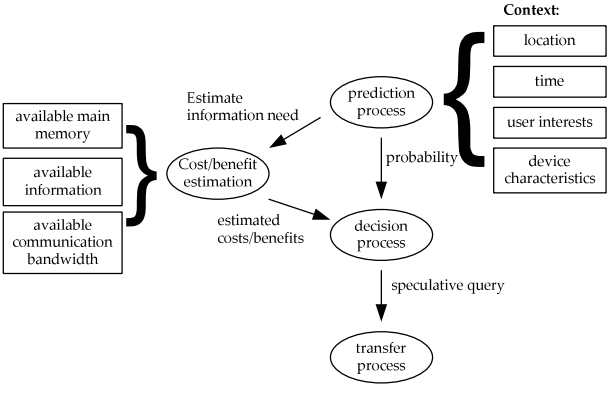


Fig. 4. Structural overview of a prefetching process

In cases where there is not enough main memory available but sufficient bandwidth, there are a few different options to react to this problem:

cache invalidation – all information which is out of the range of the user will be deleted.

take the user's interests into account – prioritize the data which is needed by the user.

minimize window size – travel time is related to a given stretch, which can be reduced to limit the PoIs.

6 Experiments and Lessons Learned

By a given set of real test data, we analyzed our approach for a typical prerecorded movement pattern (figure 5a) and let the user move through all different location models (except a and b). A speed-distance-function $tz_{dist} = 5 - 0.175 \bullet usr_{speed}$ (km) which is defined for our application area is applied to calculate the distance for target zones (see 5b).

Location model 2c has shown us that it performs similarly to 2b even when our speed distance function is applied. The major difference comes into place when bandwidth $\rightarrow 0$, because response times increases and a user might move outside a target zone, where a new query will be performed. The use of a rectangle is capturing this problem and has the advantage that it looks ahead and therefore initiates a follow up request to a later state. So the window size between requests is increased.

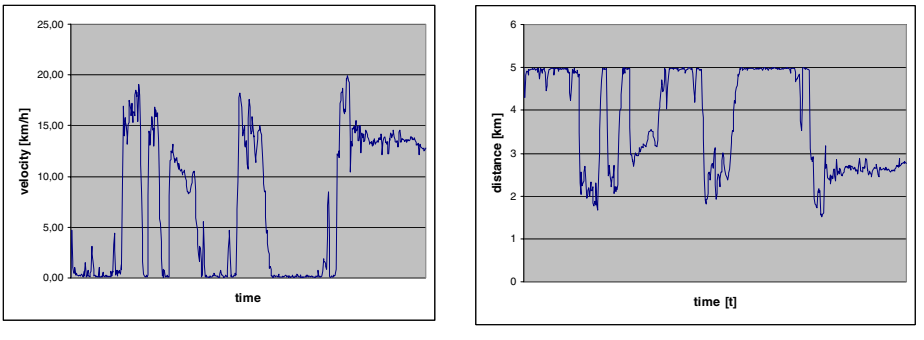


Fig. 5. a.) Typical movement pattern of a boat profile; b.) Applied speed-distance-function

The use of LACs has a slight improvement when an index is created and relevant objects are directly accessed. The problem with the previous two models appears when the direction changes dramatically, because they do not consider any topology. That's why we investigated the graph model. Our investigations have shown that the distance between two nodes should be larger than 500 m and smaller than 1500 m. The reason is that in case of 500 m there are not many PoIs which fall into such regions and in situation b the user may move further, before the system detects a closer *LocationNode* in front of him.

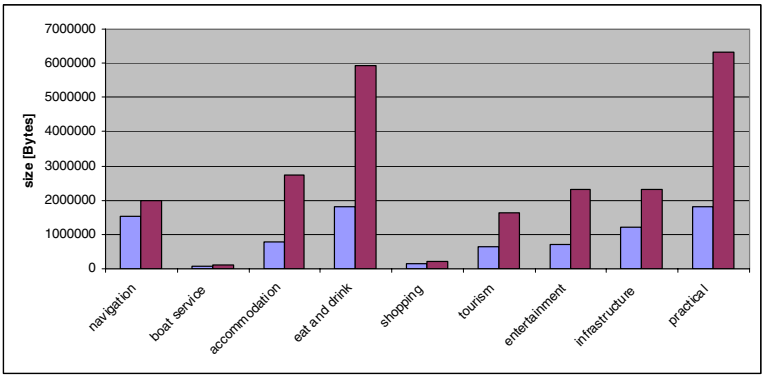


Fig. 6. Applied speed-distance-function (left columns); without distance limitation (right)

For all location models a great improvement by using a speed distance function is applied and we could, therefore, reduce the cost of main memory and bandwidth (see figure 6). Additionally, taking the user interests into account the sizes per category can be reduced more.

7 Conclusions and Future Work

In this paper, a new concept of context-aware prefetching is proposed. Users of small resource-limited mobile devices often have a need for constant up-to-date information on the move. Communication bandwidth cannot be guaranteed particularly in outdoor areas where bandwidth is often very low. Providing relevant information in time to the right place is, therefore, a major challenge. Using the location is one important parameter to find more relevant information. Our proposed location models are used to compute a target zone and to fetch relevant information for a user. Direct access to information reduces response time and increases cache hits. We also introduced a distance speed function which takes the user movement behaviour more into account and reduces network costs and traffic load.

Applications like eureauweb can benefit from prefetching and context-awareness to provide boaters with more relevant and up-to-date information in time.

One of our next steps is to extend our experiments by investigating our approach in practice and to study in particular the different roles of users. Some more work can also be done to find an optimal window size for a target zone.

References

1. Kirchner, Holger; Mahleko, Bendick; Kelly, Mike; Krummenacher, Reto and Wang, Zhou (2004). "eureauweb – An Architecture for a European Waterways Networked Information System". Conference on Information and Communication Technologies in Tourism 2004 (ENTER 2004); Cairo, Egypt, 26-**29** January 2004; Published by Springer, Wien, New York; ISBN 3-211-20669-8, pp 65-75.

2. Gitzenis, S., & N. Bambos (2002). Power-Controlled Data Prefetching/Caching in Wireless Packet Networks. Proc. IEEE INFOCOM 2002, 21st Ann. Joint Conf. IEEE Computer and Communications Societies, New York, USA: 1405-1414.
3. Ye, T., H.-A. Jacobsen & R. Katz (1998). Mobile awareness in a wide area wireless network of info-stations. Proc. Fourth Int'l Conf. Mobile Computing and Networking (MobiCom'98), Dallas, TX, USA: 109-120.
4. Bluetooth membership, www.bluetooth.org.
5. UMTS forum, www.umts-forum.org.
6. Sharifi , G., Vassileva, J., Deters, R. (2004) Seamless Communication and Access to Information for Mobile Users in a Wireless Environment, Proceedings ICEIS'2004 International Conference on Enterprise Information Systems, Porto, April 15-17, 2004.
7. Imai, N., H. Morikawa & T. Aoyama (2001). Prefetching Architecture for Hot-Spotted Network. In Proceedings of IEEE International Conference on Communications (ICC2001), pp.2006-2010, Helsinki, Finland, June 2001.
8. Kubach, U. & K. Rothermel (2001). Exploiting Location Information for Infostation-Based Hoarding. Proc. Seventh Ann. Int'l Conf. Mobile Computing and Networking (MobiCom'01), Rome, Italy: 15-27.
9. Cho, G. (2002). Using Predictive Prefetching to Improve Location Awareness of Mobile Information Service. Lecture Notes in Computer Science Vol. 2331/2002 (Int'l Conf. Computational Science (ICCS 2002), Amsterdam, The Netherlands), Springer Verlag: 1128-1136.
10. Kirchner, Holger; Krummenacher, Reto; Risse, Thomas; Edwards-May, David (2004). A Location-aware Prefetching Mechanism, Fourth International Network Conference (INC 2004), 6-8 July 2004, Plymouth, UK, ISBN 1-84102-125-3, pp. 453-460.
11. Kirchner, Holger; Glöckner, Daniel; Bieber, Gerald; Gabrecht, Sven (2005). Addressing geographic objects of unique location areas. Workshop für Ortsbezogene Anwendungen und Dienste, Universität Stuttgart, Germany June 16-17, 2005.

Location-Based Mobile Querying in Peer-to-Peer Networks

Michel Scholl[1], Marie Thilliez[2,*], and Agnès Voisard[3]

[1] Cedric, Conservatoire National des Arts et Métiers, 75141 Paris Cedex 03, France
Scholl@cnam.fr
[2] LAMIH, Université de Valenciennes, Le Mont Houy, 59313 Valenciennes Cedex 9, France
Marie.Thilliez@univ-valenciennes.fr
[3] Fraunhofer ISST and FU Berlin, Mollstr. 1, 10178 Berlin, Germany
Agnes.Voisard@isst.fhg.de

Abstract. Peer-to-peer (P2P) networks are receiving increasing attention in a variety of current applications. In this paper, we concentrate on applications where a mobile user queries peers to find either data (e.g., a list of restaurants) or services (e.g., a reservation service). We classify location-based queries in categories depending on parameters such as the user's velocity, the nature of the desired information, and the anticipated proximity of this information. We then propose query routing strategies to ensure the distributed query evaluation on different peers in the application while optimizing the device and network energy consumption.

1 Introduction

Peer-to-peer (P2P) networks are receiving increasing attention in all kinds of applications that range from music file exchange to mobile gaming. One advantage of a P2P principle of distribution is that it allows the sharing of a large quantity of information. To achieve good performance, contextual information should be taken into account. We focus here on applications where (1) information shared by a community is spread over the territory and accessible by static powerful computers (peers) spatially close to this information and (2) mobile users equipped with low energy/light computing facilities look for information close to their current location on the territory. The notion of user context usually encompasses static information such as personal profile but also dynamic information such as his or her location or speed. We here exploit the later. Besides, we utilize some characteristics of the resources to be found, which we denote the query context. In generally distributed frameworks, when a user requests information, the following two modes of interaction are considered: in the *push* mode the system sends him or her "relevant" information and in the *pull* mode the user explicitly asks for information at a certain time instant. This is the focus of our work. A user queries one or more near-by peers to find out about resources. Resources can be data (e.g., a list of restaurants) or services (e.g., a reservation service that obeys constraints like a low cost) represented by tuples in a relational database. Furthermore, each resource is geo-referenced, i.e., it has a point location in the 2D

[*] This work was done while the author was visiting Cedric lab.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 166 – 175, 2005.
© Springer-Verlag Berlin Heidelberg 2005

space - a tuple of *x* and *y* coordinates. The query is assumed to be a relational query augmented with a proximity spatio-temporal predicate (typically a range query or a nearest neighbor query) which permits to constraint the resources to be found to be in the vicinity of the user. Depending on the query and the strategy, the peers located near-by the one to which the query was issued may try to answer the query and, if necessary, will transmit it in turn to their neighbor peers as long as all these peers are in the user vicinity. Our goal is to classify location-based mobile querying and propose routing strategies in P2P networks that exploit both the environment of the users – their context - and the query context, namely: his or her user mobility materialized by his/her velocity, the type of the desired information (spatial, spatio-temporal, etc.), and the proximity of the desired information.

In distributed database querying, a query has two components which do not necessarily borrow the same path through peers:

1) **Query routing.** A query is issued by the mobile user to initial peer(s), which possibly forward(s) the query to neighbor peers.
2) **Answer routing.** Once a peer has found one answer – i.e., the result of the query is not empty - the latter has to be forwarded to the user. Depending on the strategy, the answer does not necessarily follow the same path through peers as the query itself. In the case of mobile users, the answer forwarding process has to be aware of the user motion model.

This paper is organized as follows. Section 2 gives some background on the considered P2P networks and a simple classification of location-based mobile querying. Section 3 discusses some related work. In Section 4, we specify the parameters of a query, choose three classes of queries and, for each of them, discuss and propose simple query routing strategies, considering both query and answer routing. Section 5 draws our conclusions.

2 Location-Based Mobile Querying: A Query Classification

2.1 Architecture

The resources to be shared by several participants called users and possibly mobile are distributed on several static peers (Cf. Figure 1). Peers generally correspond to robust computers. Mobile users, on the other hand, are equipped with handheld devices such as smart phones or PDAs. Peers centralize information about local resources and neighbor connected mobile users. Handheld devices are thin clients employed by users to submit their queries and consult the results. A peer is connected to another one called a *neighbor* peer in function of their networks ranges, particularly wireless networks ranges. In Figure 1, the network ranges are represented by circles.

Handheld devices can communicate not only with peers but with other handheld devices as well, as long as they are in their range. However, our architectural choices are such that a query issued by a handheld device is only evaluated on static peers and not on other handheld devices, therefore reducing communication costs. Due to this distinction of two types of devices, namely handheld devices and peers, our

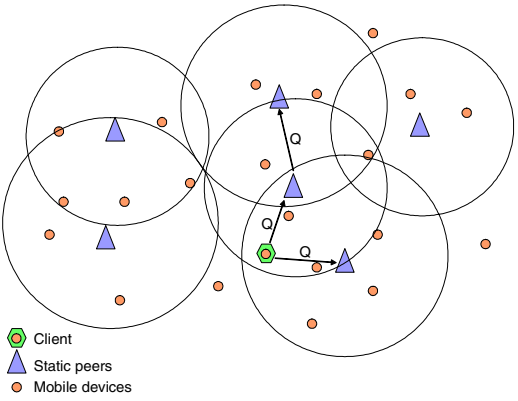


Fig. 1. Representation of the architecture

architecture is closer to the so-called hybrid Peer-To-Peer (P2P) architecture [12]. Thanks to its partial data centralization and communication cost reduction, hybrid P2P allows to improve performance and scalability.

In the following we assume that (1) information available to the users is structured according to a relational schema common to all peers, and (2) peer *p* stores tuples representing resources spatially close to the peer and possibly has some cached information about its neighbor peers and only those peers.

2.2 Parameters

When considering mobile applications and query routing strategies, many dimensions may be of interest. Let us consider three dimensions:

- The mobility of the client, which can be low or high. If the user is not moving fast or, even, is not moving, we shall say that the user is static or fixed (both terms are used interchangeably in the paper). This case typically corresponds to low network traffic and a user motion so slow that, by the time an answer reaches a user, he or she is still in the range of the peer to which the query was issued. Else we say that the user is mobile.
- The spatial (e.g., near-by drugstores), temporal (e.g., radio shows), or spatio-temporal (e.g., traffic or weather information) type of the requested information. Such a characteristic of the queried resources is important as the user who issues a query has a location and issues the query at a given time instant. Moreover, the proximity and time validity predicates in the query formulation have an impact on its answer.
- Another parameter of interest in many applications is the cardinality of the answer over the network, i.e., the possible rarity of the information. For instance, a user maybe looking for a particular friend (who is unique) or for a store that has musical scores from the XVIIIth century (*rare* and accessible only from a few peers). In contrast, gas stations are *frequent* and accessible from a large number of peers. Usually, when one is looking for a rare resource, space and time restrictions are loose or even not an issue.

Table 1 gives a query classification based on these three parameters.

Table 1. Example queries for various clients and resources and rarity of the solutions

User\Resource	Fixed		Mobile	
	Rare	**Frequent**	**Rare**	**Frequent**
Fixed	***Q1a -1:*** Find a picture of *this* monument ***Q1a- 2:*** Find a musical library with scores from the XVIIIth century	***Q1b:*** Find the picture of this monument taken by Rosa Newton	***Q3a:*** Find friend Joe	***Q3b:*** Find a taxi near my hotel
	Rare	**Frequent**	**Rare**	**Frequent**
Mobile	***Q2a:*** Find a Breton restaurant as I walk	***Q2b:*** Find a gas station as I drive	***Q4a :*** Find friend Joe as we both drive	***Q4b :*** Find a taxi as I walk

3 Related Work

Related to routing strategies in wireless and mobile P2P environment, [13,1,4] are examples of solutions to improve search in P2P networks. However, these solutions are dedicated to wired P2P network such as Internet. In our environment, the users are mobile and querying is based on the location of the information. In particular, user mobility implies that we have to consider both the dynamicity of environment due to heterogeneity, unreliability, mobility, and so on, and the limited energy of handheld devices of the mobile users.

Location-dependent queries (LDQ) [7] consider the geographic localization of mobile users in the query evaluation process. Different solutions can be used to localize the user, such as GPS [3] or wireless network based solutions [9]. Some recent studies present solutions to evaluate LDQs in mobile environments. Many LDQ evaluation solutions are dedicated to centralized environments such as cellular networks [10]. Other solutions are based on distributed environments [8], however, these solutions do not consider different mobility profiles in the query evaluation (e.g., pedestrian, car driver, and so on.). Other studies concern the evaluation of LDQs using caching techniques [2].

In [11], various challenging issues in mobile querying in a distributed environment are generally presented. Compared to this work, ours presents a classification of wireless queries and focuses on routing strategies to evaluate LDQs. Solutions in query routing for wired environments exist [5,6], however, to the best of our knowledge, none of them considers the user mobility and the energy constraints of mobile devices.

4 Query Routing Strategies

This section, proposes three routing strategies as a function of the user mobility profile and the rarity of requested information, with the following restrictions:

- We do not consider queries where a mobile user is looking for a mobile resource: all resources that a user is querying are static and associated with a unique peer.

- While there exist a variety of spatial predicates, such as: resources within a rectangle, within a disc of radius ε (range query), closest resource ("Nearest Neighbor" or NN query), or *k* closest resources (k-NN query), we restrict our attention to range queries.
- We assume that a query asks for resources to be found immediately, i.e. at the query issue time. Queries whose result is to be found in the future lead to other strategies which take into account the anticipation of the query. We leave them as a future work.
- In the following, we omit querying resources that are only valid in some time intervals or at some time instants. We believe that the extension of the proposed strategies to temporal (news) and spatio-temporal predicates (near-by sportive events, theater plays) taking into account the temporal type of the resource is straightforward.

A query is specified by the following tuple: $Q=[id_c, id_q, sql, \varepsilon, t_0, t_{max}, traj]$.

Query Q with identifier id_q, issued at time t_0 by a mobile client c with identifier id_c and trajectory *traj* is interpreted as: "Return all resources satisfying query *sql* within a disc of radius ε, prior to time t_{max}".

Peer p chooses a strategy that depends on the rarity of the resource (we assume that rarity is an attribute of one of the associated relations, but it can be inferred by the system as well), t_{max} and the speed of the client whose components are specified in parameter *traj*. Depending on the speed of c, and the rarity of the resource, one out of a number of strategies is chosen. We consider the three following cases:

- ***fixed-to-fixed (FtoF)*:** the client speed is low, t_{max} is high, and the traffic is light enough so as to guarantee that peer p has enough time to find the answer(s) (among its resources or by its neighbor peers) and forward it back to client c which, in the meantime, has not left zone Z controlled by p. Q1* (Table 1) is an example of ***FtoF*** query.
- ***mobile-rare (MR)*:** the speed/time max/traffic configuration is such that the client is likely to have left zone Z at time $t \text{ in }]t_0,t_{max}]$, once peer p - which has an answer a - wants to forward it back to c. One typical consideration would be: high value of t_{max} and high speed. Since they are rare, all answers found must be forwarded to the client wherever he or she stands at time $t'>t$. If the found resource is itself mobile, then its trajectory at time t is forwarded as well to c. In Table 1, there are 2 examples of ***MR*** queries : Q2a or Q4a.
- ***mobile-frequent (MF)*:** as for the MR case, the client has a good chance for having left the zone Z, however, here it is not worth forwarding the query since the searched resource has a large probability of being found in other peers along c trajectory. The simple strategy chosen here is to forward the query to peers close to c trajectory; if peer p' at time t has an answer it sends it back to c if c is in peer p' zone at time t. Queries Q2b and Q4b (Table 1) are ***MF*** queries.

Although there exist several strategies for each of the aforementioned scenarios, we give an algorithm for only one in each case. It is further assumed that each peer p locally stores the distance to its furthest resource d_{max} as well as to each of its neighbor peers p' the furthest resource distance d_{max}' to p'. A refined variant which would require more space would be to store the maximum distance to peer p (same for its neighbors) for each of the relations. No further assumption is made on the zones

covered by a peer. In each peer, *d(p)* stores for peer *p* the maximum distance to any resource registered in *p* (*p* is either the peer hosting the table or any of its neighbors). *distance(a,b)* denotes the Euclidean distance between two points *a* and *b*.

4.1 Strategy FtoF

The strategy (functions *FtoFquery* and *FtoFanswer*) works as follows. *Q* is issued by *c* to peer p_0 (*c* belongs to the zone associated with p_0). If *distance(c,*p_0*)* is larger than $\varepsilon + d(p_0)$ then no resource under *p* satisfies the query. Else *p* runs the query and checks for each answer *a* under its zone, whether its (Euclidean) distance to *c* is less than ε. It also sends query *q* to its neighbor peers *p'* whose distance to *c* is less than $\varepsilon + d(p')$ and waits for its answers. If a neighbor *p'* has in turn another neighbor peer *p"* such that $distance(c,p") \leq \varepsilon + d(p")$, then query Q is forwarded to *p"*. *p"* answers if any are returned to *p'*, etc. Depending on the traffic, a variant strategy would choose to send back the answer directly to the initial peer p_0. Note that in the applications we target, in which resources are to be looked for in the vicinity of the client, the expected length of the chain of visited peers should be small (one or two). Eventually, peer p_0 has collected all answers and sends them back to the client. *FtoFanswer(A,p)* forwards the answers *A* back to peer *p*. The initial call is *FtoFquery(Q,*t_0*,c,*p_0*)*.

```
1:   Algorithm FtoFquery (Q,t,p,p')
2:   Input: Q=[ id_c , id_q , sql , ε , t_0 , t_max , traj , ],
3:            p'=current peer (to which the query has been sent)
4:            p=The peer that sent the query
5:   Output: A= set of answer tuples
6:   A:= emptyset
7:   If  t>t_max then FtoFanswer(A,p)
8:   Else if distance(c,p') > ε +d(p') then FtoFanswer(A,p)
9:        Else For each a in sql
10:                If distance(a, c) ≤ ε then A+={a}
11:       For each neighbor peer p", A+={FtoFquery(Q, t, p', p")}
12:       FtoFanswer(A,p)
```

4.2 Strategy MR

Strategy MR works as follows. *Q* is issued by *c* like in strategy FtoF (Function *MRQuery*). If the current peer does not have an answer, Q is forwarded to its neighbor peers. Else each peer p having found a resource at time $t_1 < t_{max}$ forwards it back according to the following strategy (algorithm *MRanswer*). Given the trajectory, *p* estimates the client location $l_\theta = [x,y]$ after a time θ. θ is an estimator of the time necessary to reach the mobile client. θ depends on the client speed *s*, the level of traffic, and the distance *d* between the peer and the client at time t_1. A simple estimator would be $\theta = k d^{\gamma} s^{\beta}$ where *k* depends on the traffic. Let $\Delta(p,p')$ be the line between *p* and a neighbor peer. $\Delta(p,l_\theta)$ is the line between p and the mobile client at time θ. Let $\alpha\{ \Delta 1, \Delta 2 \}$ denote the angle between the 2 lines in argument. Then peer *p* chooses as a neighbor peer (*p'*) to which the query is to be forwarded the one such that $\alpha\{\Delta(p,p'), \Delta(p,l_\theta)\}$is minimum. Then when *p'* receives the answer, either the client is inside the zone covered by *p'*, and the answer is forwarded back to the client,

```
1:  Algorithm MRquery (Q,t,p')
2:  Input: Q=[ id_c , id_q , sql , ε , t_0 , t_max , traj , ],
3:        p'=current peer (to which the query has been sent)
4:  Output: A= set of answer tuples
5:  A:= emptyset
6:  If  t>t_max then Return
7:  Else if distance(c,p') > ε +d(p') then Return
8:      Else For each a in sql
9:          If distance(a ,c) ≤ ε then A+={a}
10:     If A non empty then begin
11:                 min:=360; θ =kd^γ s^β; l_θ := estim(traj,t_0, s, θ)
12:                 For each neighbor peer p if α{Δ(p, p'), Δ(p', l_θ )}<min
13:                                 then begin
14:                                     min = α{ Δ(p, p'), Δ(p, l_θ )}
15:                                     p_1:=p
16:                                     end
17:                 MRanswer(A,p_1,Q)
18:                 end
19:                Else For each neighbor peer p", A+={MRquery(Q,t,p")}
```

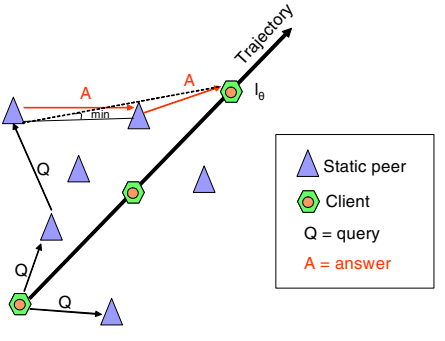


Fig. 2. Query and answer routing in strategy MR

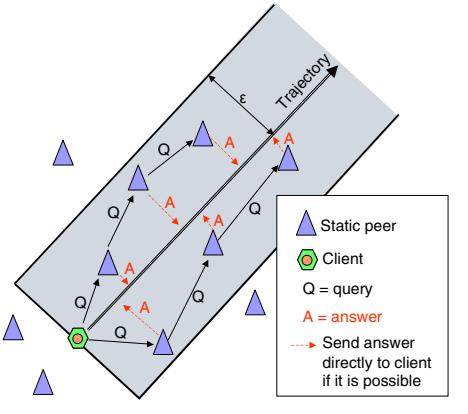


Fig. 3. Query and answer routing in strategy MF

```
1:  Algorithm MRanswer(A,p',Q)
2:  If t>t_max then return
3:      Else if  c in zone(p) then return (A,c)
4:          Else begin
5:              min:=360; θ = kd^γ s^β; l_θ := estim(traj,t_0, s, θ)
6:              For each neighbor peer p if α{ Δ(p,p'), Δ(p',l_θ )}<min
7:                                  then begin
8:                                      min = α{ Δ(p,p'), Δ(p,l_θ )}
9:                                     p_1:=p
10:                                    end
11:             MRanswer(A,p_1,Q)
12:             end
13: Return
```

or the answer has to be forwarded to *p"*. *p"* is chosen according to the same algorithm. However, we have two variants: either compute the position according to the same value of θ or refresh θ and thus the future location of the client. The initial call is *MRquery* (Q,t_0,p_0). Function *estim* estimates $l_{\theta.}$

4.3 Strategy MF

The resources are found as in *MRQuery* (function *MFquery*). However, in contrast to strategy MR, possible results are numerous and ubiquitous. Then, one minimizes traffic and energy consumption related to communications between peers by not forwarding results if the client is out of the spatial scope. The searched resources can be found later on, on the mobile user trajectory. In the following variant, if peer *p* has an answer and if client *c* is still in the spatial range of *p*, then the answer is forwarded back to *c*. Else (*c* not in range anymore) the query is forwarded to neighbor peers close to c trajectory. Function *inScope* in Algorithm *MFQuery* returns true if the peer specified as a parameter is in the semi-rectangle illustrated in Figure 3. This semi-rectangle represents the spatial scope of the query as a function of the client trajectory and ε. In another "continuous query", to obtain all possible results, variant peer *p* would forward the query to neighbor peers anyway (even if the answer was successfully sent back to *c*). A third variant would be to choose a strategy similar to that of *MRanswer* for forwarding the query more adapted to cases where the client is far away once a query in a peer has failed.

```
1:  Algorithm MFquery (Q,t,p')
2:  Input: Q=[ id_c , id_q , sql , ε , t_0 , t_max , traj , ],
3:          p'=current peer (to which the query has been sent)
4:  Output: A= set of answer tuples
5:  A:= emptyset
6:  If t>t_max then Return
7:    Else begin
8:         if distance(c,p') < ε +d(p') then
9:            begin
10:            For each a in sql
11:                 If distance(c,a) ≤ ε then A+={a}
12:            M := false;
13:            If A non empty then M := MFanswer(A,p",Q)
14:            If (M= false and t< t_max) then
15:               For each neighbor peer p",
16:                  If inScope(p", traj, t, ε) then MRquery(Q,t,p")
17:            end
18:         end
```

```
1:  Algorithm MFanswer (Q,t,p')
2:  M := false
3:  If t>t_max then return (M)
4:     Else If c in zone(p) then
5:                    begin
6:                      Forward (A,c); M:=true; return (M)
7:                    end
8:                   Else return(M)
```

5 Conclusion

In this paper, we presented a simple classification of location-based mobile queries in P2P networks according to context variables of both the user and the query, namely the client mobility, the spatio-temporal nature of the query, and its rarity. These parameters have an impact on both query routing and answer routing. We chose three classes of queries and described for each of them a detailed query and answer routing strategy. As a future work we intend to evaluate the performance of these strategies by comparing them to other variants among which some were suggested earlier. Among the parameters that impact on the performance, the level of traffic (number of clients and number of queries) as well as the client speed are noteworthy.

Acknowledgments. We wish to thank the anonymous reviewers for their comments. This work was carried out in the framework of a PROCOPE French-German grant.

References

[1] M. Demirbas, H. Ferhatosmanoglu. Peer-To-Peer Spatial Queries in Sensor Networks. In *Proc. 3rd Intl. IEEE Conference on Peer-to-Peer Computing (P2P'03),* Computer Society Press, Los Alamitos, CA, USA, 2003.

[2] T. Doulkeridis, V. Zafeiris, and M. Vazirgiannis. The Role of Caching in P2P Service Discovery. In *Proc. Intl. ACM Conference on Mobile Data Management (MDM)*, ACM Press, New York, N.-Y, 2005.

[3] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning Systems: Theory and Practice*, *5th edition*. Springer Verlag, Berlin/Heidelberg/New York, 2001.

[4] S. K. Goel, M. Singh, D. Xu, B. Li. Efficient Peer-To-Peer Data Dissemination in Mobile Ad-Hoc Networks. In *Proc. Intl. Workshop on Ad Hoc Networking (IWAHN, in conjunction with Intl. Proc. Intl. Conference on Parallel Processing, ICPP),* 2002.

[5] M. Iles, D. Deugo. A Search for Routing Strategies in a Peer-To-Peer Network Using Genetic Programming. In *Proc. IEEE Symposium on Reliable Distributed Systems (SRDS)*, Computer Society Press, Los Alamitos, CA, USA. 2003.

[6] E. Michlmayr, S. Graf, W. Siberski, W. Nejdl. Query Routing with Ants. In *The Semantic Web: Research and Applications, Second European Semantic Web Conference*, A. Gomez-Perez, J. Euzenat (Eds.), Lecture Notes in Computer Science No. 3532, Springer Verlag, Berlin/Heidelberg/New York, 2005.

[7] J. Schiller, A. Voisard. *Location-Based Services*. Morgan Kaufmann/Elsevier, San Francisco, CA, USA, 2004.

[8] M. Thilliez and T. Delot. Evaluating Location Dependent Queries Using ISLANDS. In *Advanced Distributed Systems: Third International School and Symposium, (ISSADS)*, Lecture Notes in Computer Science No. 3061, Springer Verlag, Berlin/Heidelberg/New York, 2004.

[9] M. Thilliez, T. Delot, S. Lecomte. An Original Positioning Solution to Evaluate Location-Dependent Queries in Wireless Environments. *Journal of Digital Information Management - Special Issue on Distributed Data Management*, (3) 2, 2005.

[10] J. Xu, D.K. Lee. Querying Location-dependent Data in Wireless Cellular Environment. In *Proc. W3C Workshop on Position Dependent Information Services*, 2000.

[11] B. Xu, O. Wolfson. Data Management in Mobile Peer-To-Peer Networks. In *Proc. of the 2nd Intl. Workshop on Databases, Information Systems, and Peer-to-Peer Computing (DBISP2P'04),* Lecture Notes in Computer Science No. 3367, Springer Verlag, Berlin/Heidelberg/New York, 2004.

[12] B. Yang, H. Garcia-Molina. Comparing Hybrid Peer-To-Peer Systems. In *Proc. of the 27th Int. Conf. On Very Large Data Bases*, Morgan Kaufmann/Elsevier, San Francisco, CA, USA. 2001.

[13] B. Yang, H. Garcia-Molina. Improving Search in Peer-To-Peer Networks. In *Proc of IEEE Intl. Conference on Distributed Computing Systems (ICDCS)*, Computer Society Press, Los Alamitos, CA, USA, 2002.

Seamless Engineering of Location-Aware Services*

Gustavo Rossi[1,2], Silvia Gordillo[1,3], and Andrés Fortier[1]

[1] LIFIA, Facultad de Informática, UNLP, La Plata, Argentina
[2] CONICET
[3] CICPBA
{gustavo, gordillo, andres}@lifia.info.unlp.edu.ar

Abstract. In this paper we present a novel approach to design and implement applications that provide location-aware services. We show how a clear separation of design concerns (e.g. applicative, context-specific, etc) helps to improve modularity. We stress that by using dependency mechanism among outstanding components we can get rid of explicit rule-based approach thus simplifying evolution and maintenance. We first discuss some related work in this field. Next, we introduce a simple exemplary scenario and present the big picture of our architectural approach. Then we detail the process of service definition and activation. A discussion on communication and composition mechanisms is next presented and we end presenting some concluding remarks and further work.

1 Introduction

Building applications that provide location-aware services (i.e. those services which depend on the user position) is usually hard. Moreover, maintenance and evolution of this kind of software is even harder. There are many reasons for this:

- Dealing with location (and other kind of context) information is hard because this information has to be acquired from non-traditional devices and must be abstracted to make sense to applications [4].
- Applications usually evolve "organically" [1]; new services are added, location models change, new physical areas are "tagged" to provide services.
- Application objects contain location (or contextual) information (e.g. a room is located in some place in a building) which is usually not clearly decoupled from other application-specific information (e.g. courses given in the room). As a consequence it is difficult to engineer these concerns separately.
- Adapting to context requires different engineering techniques many of which are usually misunderstood. For example, most approaches use the rule-based paradigm to express adaptation policies, such as: "When being in a room,

* This paper has been partially supported by the Argentine Secretary of Science and Technology (SeCyT) under the project PICT 13623.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 176–185, 2005.
© Springer-Verlag Berlin Heidelberg 2005

provide services A, B and C". While rules can be often useful (especially when we want to give the user the control of building his own rules), many times rule sets might become too large and thus may be necessary to use more elaborated design structures to improve maintenance and evolution.

Our approach is based on a clear separation of concerns that allows us not only to decouple context sensing and acquisition (as in [12]), but mainly to improve separation of inner application modules, easing extension and maintenance. To achieve this, we make an extensive use of dependency (i.e. subscribe/notify) mechanisms to relate services with locations. In our approach, services are "attached" to locations in such a way that a change in the user's location triggers the necessary behavior to display the available services. Further use of the same philosophy may allow applying it to other contextual information, such as time and role.

The main contributions of our paper are the following:

- We show how to separate application concerns related with context awareness to improve modularity. Also, as a by-product, we indicate a strategy to extend legacy applications to provide location and other context-aware services. A concrete architecture supporting this approach is presented.
- We show how to objectify services and make them dependent of changes of context; in particular we emphasize how to provide location-aware services.

The rest of the paper is organized as follows: In Section 2 we briefly discuss related work. In Section 3, we introduce a simple motivating example both to present the problems and to use it throughout the paper; In Section 4 we describe the overall structure of our architecture. In Section 5 we focus on service specification and activation. We discuss some further issues in Section 6 and present our concluding remarks in Section 7.

2 Related Work

The Context Toolkit [4] is one of the first architectural approaches in which sensing, interpretation and use of context information is clearly decoupled by using a variant of the MVC architectural style [9]. Meanwhile, the Hydrogen approach [7] introduces some improvements to the capture, interpretation and delivery of context information with respect to the seminal work of the Context Toolkit. However, both in [4] and [7] there are no cues about how application objects should be structured to seamlessly interact with the sensing layers. As shown in Section 4, our approach proposes a clear separation of concerns between those object features (attributes and behaviors) that are "context-free", those that involve information that is context-sensitive (like location and time) and the context-aware services. By clearly decoupling these aspects in separated layers, we obtain applications in which modifications in one layer barely impact in others. The idea of attaching services to places has been used in [8], though our use of dependency mechanisms improves evolution and modularity following the Observer's style [6].

In [5], Dourish proposes a phenomenological view of context. In this work, context is considered as an emergent of the relationships and interactions of the entities involved in a given situation. This idea improves existing approaches, in which context is viewed as a collection of data that can be specified at design time and whose structure is supposed to remain unaltered during the lifetime of the application. Similarly to [5], we don't treat context as plain data on which rules or functions act, but as the result of the interaction between objects, each one modeling a given context concern. In addition, we do not assume a fixed context shape, and even allow run-time changes on the context model.

3 An Exemplar Scenario

Future location-aware systems will not be built from scratch but rather as the evolution of existing legacy systems. Suppose for example, that we want to enhance a university information system with context-aware functionality to make it work as the example in [13]. Our system already provides information about careers, courses, professors, courses' material, time-tables, research projects, etc. and is used (e.g. with a Web interface) by professors, students and the administrative staff.

In our enhanced (location-aware) system, when a student arrives to the Campus he can invoke some general services; when he enters a room he is presented with those services that are meaningful for that room (while still having access to the former services); for example, he can get the corresponding course's material or obtain information about the course, its professor, etc. When he moves to the sport area, he can query about upcoming sport events and so forth. Other kinds of users (professors) will have access to a (partially) different set of services.

In this paper we show how we solved one of the most challenging design problems in the scenario: how to seamlessly extend our application in order to be location-aware, i.e. to provide services that correspond to the actual location context. For the sake of conciseness we barely mention other kinds of context-aware adaptation.

4 Our Architectural Approach. An Outline

To cope with the dynamic and evolving requirements of this kind of software we devised a layered architecture in which we combine typical inter-layer communication styles [3], with the dependency mechanisms of the Observer design pattern [6]. In Figure 1 we present a high level view of the most important modules of the architecture shown as UML packages; we next describe each of them in detail together with the main communication mechanisms, in the context of the example.

4.1 Architectural Layers

In the **Application Model** we specify the application classes with their "standard" behaviors; application classes and methods are not aware of the user's

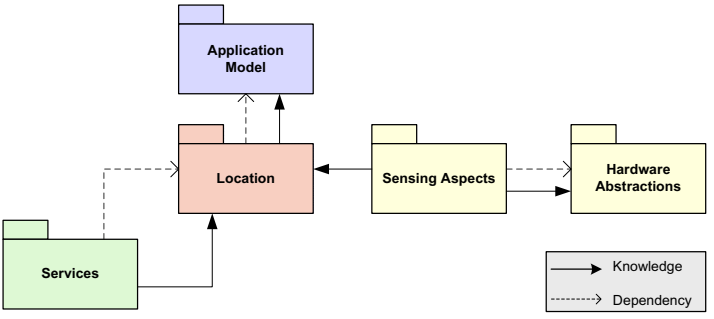


Fig. 1. The Architecture of a Location-Aware Application

context, neither they exhibit location (or other context-related) information. In our example we would have classes to handle room reservations, time-table scheduling, professors and material associated with each course, etc.

In the **Location Model**, we design components that "add" location properties to those application objects that must "react" when the user is in their vicinity. For example, to be able to say that a user is in room A we first need to create a location abstraction of the corresponding room object; we clearly separate the room from the object which considers the room as a spatial object. The location layer also comprises classes for "pure" location concepts, for example corridors, maps, connecting paths, etc., which do not have a counterpart in the application layer. In our example, we may be interested in representing a map of the university building, where we find rooms that are connected by corridors. The actual user location, which represents one of the user context concerns, is also modeled in this Layer. The design decision underlying the Location layer

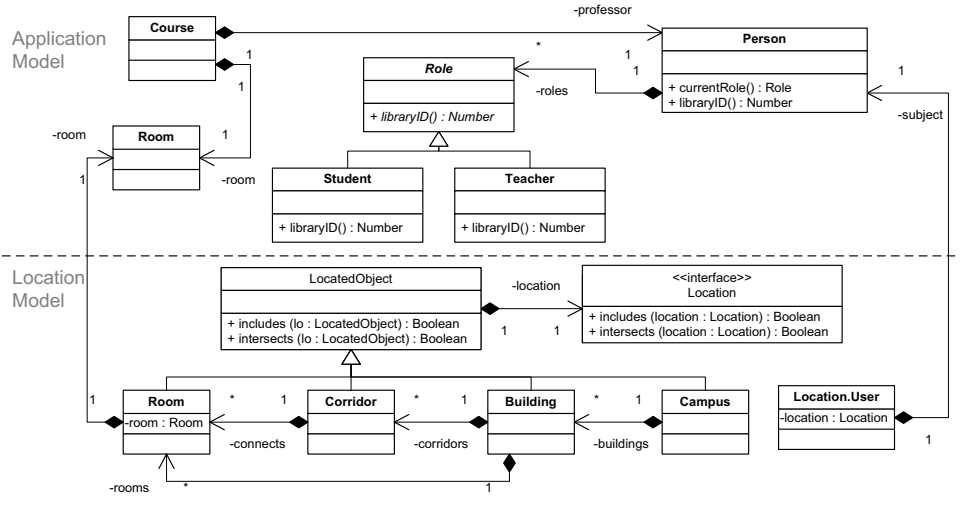


Fig. 2. Location Model vs. Application Model

(separating an object from its location in such a way that the object knows nothing about its spatial properties) is one of the keys to support seamless extension. Decoupling location objects from other application objects allows us to deal with different location models such as symbolic or geometric [10] transparently. In the Location layer, we model location as an interface, so that many different location models can be used interchangeably. In Figure 2 we show an example of the Location layer including the location of some application objects and the user location. The Location layer can be generalized into a Context layer by including time, role, activity, and other contextual information that we do not describe in this paper for the sake of conciseness.

As discussed in the next paragraph, new instances of Location can be built "opportunistically" to create service areas, i.e. those areas to which a set of services apply (e.g. one might have services that are meaningful in an area covered by the union of a corridor and a set of rooms, even though we do not consider the area as a full-fledged LocationObject).

The **Service** layer contains the (location-aware) services. These services are modeled as objects that will be further associated to certain geographic areas (that belong to the location layer) by means of a subscription mechanism. In this layer we also include an object that manages the actual user services (in particular service subscription and activation) and coordinates other user's aspects. The User object (or Service.User) plays a similar role as the Context component in [4]. A high level description of the Service Layer and the relationships with objects in the Location model are shown in Figure 3. It is important to notice the way in which we express that a service is available in a certain area: since we don't want to clutter the location layer with services stuff, the logical relationship between services and physical areas is expressed by the concept of a service area. A service area class is used at the service level; an object of this class knows which physical area it covers, by means of the location relationship (shown in Figure 3); its main responsibility is to know which services are currently available (in the area) in such a way that, when a user is standing in a location included in the service area, the new available services are presented to the user (see section 5). A service area thus, acts as a Mediator [6] between a set of services and the area in which they are available.

In the **Hardware Abstractions** layer we find the components used for gathering data, such as IButton, InfraredPort, GPSSensor, and so on; these abstractions are similar to Dey's Widget components [4].

The **Sensing** layer comprises those higher level aspects that plug the lower level sensing mechanisms (in the hardware abstraction layer), with those aspects that are relevant to the application's context and that have to be sensed. This decoupling (which could be considered an improvement of Dey's interpreters [4]) guarantees that the location model and the sensing mechanisms can evolve independently. For example, we can use a symbolic location model [10] to describe locations, and infrared beacons as sensing hardware; we can later change to a non-contact iButton seamlessly (by hiding this evolution in the sensing layer). For the sake of conciseness we do not explain these (lower-level) layers in detail.

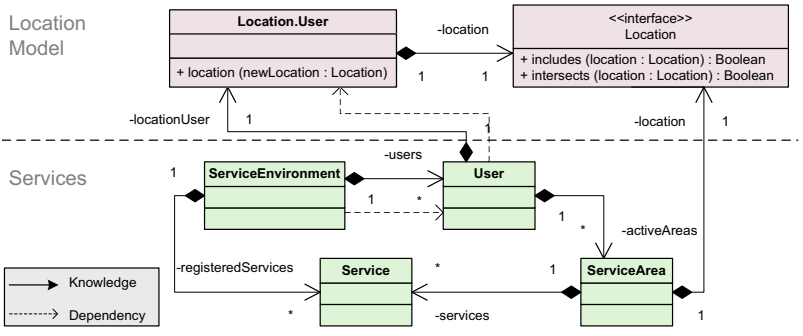


Fig. 3. The Service Layer

4.2 Communication Mechanisms

Relationships among objects in different layers follow two different styles: typical knowledge relationships (such as the relationship between the object containing a room's location and the room itself that resembles the Decorator pattern [6]) and dependency relationships (in the style of the Observer pattern [6]) that allow broadcasting changes of an object to its dependent objects.

A refinement of Figure 1 showing some of the outstanding dependency relationships is presented in Figure 4. When a change is detected in the Hardware Abstractions Layer (e.g. by an IR Port), a Location Aspect object is notified and it sends a message to the corresponding Location.User object (in the Location Model). The dependency mechanism between Location and Services allows notifying the User object which in turn notifies the Service Environment. This chain of events allows that the services corresponding to the actual user's position are

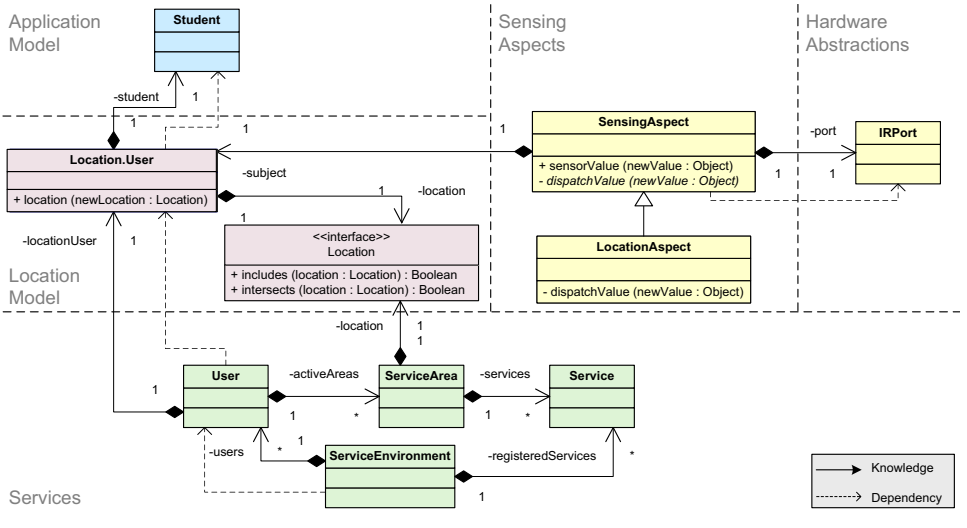


Fig. 4. Examples of knowledge and dependency mechanisms

made available to the User object which finally presents them to the actual user. In the next section, we elaborate our strategy to specify and manage services.

5 Service Specification and Activation

We consider (context-aware) services as possible independent artifacts which are developed individually and do not need to interact with each other (in fact they might even ignore that there are other services running). We also view them as extending some existing application behavior and thus they might need to interact with application objects. In this sense, even though being "isolated", they can be considered as parts of the application behavior (in fact a way to trigger application behavior).

Service users are immersed in a service environment (which in fact reifies the real-world environment). The Service.User object (i.e. the user considered from a services point of view) knows those services to which the corresponding user is subscribed to, and which services are currently available. The service environment is in turn responsible for handling available services, configuring service areas and mediating between users and services.

Services are modeled as first-class objects; this allows our framework to treat them uniformly and simplify the addition of new services.

5.1 Creating New Services

New services are defined as subclasses of the abstract class Service (Figure 4), and thus a specific instantiation of a service is an object that plays the role of a Command [6]. The specific service's behavior is defined by overriding appropriate methods of Service such as *start()* (used to perform initialization stuff), *activate()* (triggered when the users selects the service form the available services list), etc. For example, the CourseMaterial service is defined as a subclass of Service, and the message *activate()* is redefined so that a graphical interface is opened to display the course material.

5.2 Subscribing to Services

Users can access the set of available services (in an area) and decide to subscribe (or unsubscribe) to any of them. In our model, the service environment knows which services are registered, and therefore the users can query the environment for existing services in order to subscribe to them. The details of the subscription mechanism are beyond the scope of this paper; however, to mention some few interesting aspects in our framework, once a user is subscribed to a service, he can customize it (for example, the service can indicate its availability by playing a sound or by other means), or he can define additional context-aware constraints over the service (such as allowing or disallowing activation in certain situations).

5.3 Attaching Services to Spatial Areas

To provide location-awareness and to avoid the use of large rule sets, services are associated with (registered to) specific areas, called service areas. When the

user enters into a service area, all services registered to the area (to which the user has subscribed) are made available. Service areas are defined to achieve independence from the sensing mechanism, i.e. they do not correspond to the scope of a sensing device (e.g. a beacon) but to logical areas. These logical areas can be specified programmatically or interactively; they can be obtained by applying set operators to existing areas (rooms, corridors, etc) or defined arbitrarily in terms of a location model.

As an example, suppose that we want to offer a location service in which a map appears showing where the user is standing. Clearly, we would expect this service to be available in the university building or even in the entire campus. If we are working with symbolic location we would probably have a 'Building' location that is the parent of the location tree that encompasses the rooms inside the building. So, in order to provide the location service for the building, we would create a new service area that is associated with the 'Building' location (see Figure 3); with this configuration, when the user enters the building (and as long as he is inside of it) he will have that service available. Now suppose that we would like to extend this service to the entire campus; using our approach we would just need to change the area covered by the service area, which in case of symbolic location means changing the location 'Building' to 'University Campus'. Similarly, if we want to add new services to that area, we do it by adding a service to the list of services known by the service area.

5.4 Service Activation

As explained abstractly in 4.2, when the user's movement is captured by a sensor it triggers a set of messages; concretely, it sends the *location (newLocation)* message to the Location.User corresponding to the actual user. This message triggers a change in the location model that is captured (by means of the dependency mechanism) by the User object in the service layer. This object interacts

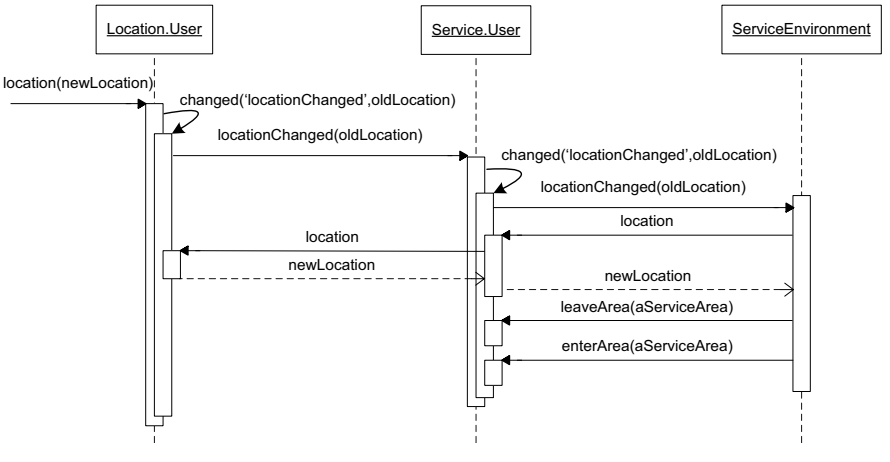


Fig. 5. Changing the Set of Active Services

with its environment to calculate, if the user entered or left a service area, and the corresponding services are made available to the user, according to his subscriptions. As mentioned before, a service is presented to a user if it is available in a service area and if the user is subscribed to it. A subset of these interactions is shown in Figure 5 by means of a UML sequence diagram.

6 Discussion

The impact of the previously described architectural style and communication mechanisms in the application's structure is somewhat evident. Separating hardware and sensing abstractions allows maintaining the benefits of well-known frameworks for context-awareness (such as [7,12]). We introduced two additional layers (Location and Service) which refine the Application Model by clearly separating contextual attributes (such as position), low-level models (such as the actual location model) and context-aware behaviors, expressed as service objects.

Viewing Services as objects allows simplifying evolution; new services are added by just creating new classes, associating them to the service environment and to specific service areas, and publishing them to allow users' subscriptions. Finally the communication structure, expressed with a dependency mechanism that generalizes the Observer pattern, provides a seamless way to activate/deactivate services. Once associated with a spatial area, which can be done either programmatically (e.g. sending a message) or interactively, the Service Environment calculates which services can be accessed by a user by just reacting to a chain of notification events. In this way we get rid of large rule sets thus simplifying addition, deletion or upgrade of services (for example to associate them to different areas).

7 Concluding Remarks and Further Work

We have presented a novel approach for designing location aware services which uses dependency mechanisms to connect locations, services and application objects. We have also shown how we improved separation of different design concerns, such as applicative, spatial, sensing, etc. We have built a proof of concept of our architectural framework using a pure object oriented environment (VisualWorks Smalltalk). By using native reflection and dependency mechanisms we easily implement the architectural abstractions shown in the paper. We used HP iPaq 2210 PDAs as user devices; location sensing was performed using infrared beacons which can be adapted to provide ids or even URLs as their semantic locations [11]. Our approach represents a step forward with respect to existing approaches in which context information is treated as plain data that has to be queried to provide adaptive behaviors.

We are now working on the definition of a composite location system that allows symbolic and geometric location models to coexist seamlessly. We are also planning to enhance the simple dependency mechanism to a complete event-based approach, delegating specific behavior to events and improving at the same

time the framework's reusability. We are additionally researching on interface aspects to improve presentation of large number of services and service maps.

References

1. Abowd, G.: Software Engineering Issues for Ubiquitous Computing. Proceedings of the International Conference on Software Engineering (ICSE 99), ACM Press, 1999, pp. 75-84.
2. Beck, K., Johnson, R.: Patterns generate architecture. Proceedings of the European Conference on Object-Oriented Programming, Ecoop '94 Lecture Notes in Computer Science.
3. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M.: Pattern-Oriented Software Architecture. John Wiley, 1996.
4. Dey, A.: Providing Architectural Support for Building Context-Aware Applications. PHD, Thesis, Georgia Institute of Technology, USA, 2001.
5. Dourish, P.: What we talk about when we talk about context. Personal and Ubiquitous Computing 8, 1 (2004) 19-30.
6. Gamma,R., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison Wesley, 1995.
7. Hofer, T., Pichler, M., Leonhartsberger, G., Schwinger, W., Altmann, J.: Context-Awareness on Mobile Devices - The Hydrogen Approach. Proceedings of the International Hawaiian Conference on System Science (HICSS-36), Minitrack on Mobile Distributed Information Systems, Waikoloa, Big Island, Hawaii, January 2003.
8. Kanter, T.: Attaching Context-Aware Services to Moving Locations. IEEE Internet Computing V.7, N.2, pp 43-51, 2003.
9. Krasner, G., Pope, S.: A Cookbook for Using Model-View-Controller User Interface Paradigm in Smalltalk-80, Journal of Object Oriented Programming, August/September, 1988, 26-49.
10. Leonhardt, U.: Supporting Location-Awareness in Open Distributed Systems. Ph.D. Thesis, Dept. of Computing, Imperial College London, May 1998.
11. Pradhan, S.: Semantic Location. Personal Technologies, vol 4(4), 2000, pp. 213-216.
12. Salber, D., Dey, A., Abowd, G.: The Context Toolkit: Aiding the Development of Context-Enabled Applications. Proceedings of ACM CHI 1999, pp 434-441.
13. Sousa, J.P., Garlan, D.: Aura: an Architectural Framework for User Mobility in Ubiquitous Computing Environments. (3rd IEEE/IFIP Conference on Software Architecture) WICSA 2002: 29-43.

Context-Aware Negotiation for Reconfigurable Resources with Handheld Devices

Timothy O'Sullivan and Richard Studdert

Computer Science Department, University College Cork, Ireland
{t.osullivan, r.studdert}@cs.ucc.ie

Abstract. Next-generation handhelds are expected to be multi-functional devices capable of executing a broad range of compute-intensive applications. These handheld devices are constraint by their physical size, their computational power and their networking ability. These factors could hinder future performance and versatility of handheld devices. Reconfigurable hardware incorporated within distributed servers can help address portable device constraints. This paper proposes a context-based negotiation and bidding technique enabling a handheld device to intelligently utilise its surrounding reconfigurable resources. The negotiation protocol enables handhelds to optimally offload their computational task. The contextual aspect of the protocol uses the location of a mobile device to identify the urgency of a user request. This helps to optimise the quality of service experienced by a handheld user. An architectural framework currently being deployed and tested as well as overall future objectives are outlined.

1 Introduction

Handheld devices were traditionally designed and optimised to perform a specific task. In contrast, current and future generations of handhelds are evolving and are expected to deliver increasingly diverse functionality. Portability has remained an essential characteristic for handhelds. Manufacturing devices to be portable places limitations on their physical size, computational power and networking ability. These constraints hinder the ability of a handheld to execute a broad range of applications.

This raises serious performance and versatility issues for next-generation handhelds. These concerns can be addressed with the development of innovative design and deployment methodologies to enable next-generation portable devices meet their expectations. This work proposes integrating networked reconfigurable hardware resources into the environment of a handheld device. Field Programmable Gate Array (FPGA) hardware is the enabling technology of these reconfigurable resources. Adaptive servers that integrate this technology have the ability to perform mobile device computational requests in hardware. This hardware execution increases adaptive server performance. The adaptive server still retains much of the flexibility of a software solution due to the reconfigurable aspect of the FPGA technology [1]. Distributed adaptive servers are capable of significantly enhancing the performance and versatility of handhelds [2].

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 186 – 195, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Portable devices require a sophisticated middleware framework to enable them to effectively utilise these networked reconfigurable resources. An agent-based middleware is an ideal environment for mobile device management. Agents are efficient in their use of bandwidth and can deal with intermittent network connections. They are also capable of coherently handling the execution and transmission of reconfigurable hardware-software based computations. An agent can effectively represent, communicate and work towards a user's interests.

This paper proposes a context-based negotiation strategy within an agent based framework. The framework enables intelligent utilisation of surrounding reconfigurable resources by mobile devices.

A context-aware handheld can proactively assess its environment. The information gathered from this assessment can better inform the decision-making process of agents operating within adaptive servers with regard to resource allocation. The execution of the offloading protocol is influenced by the location of the portable device. This contextual information enables an adaptive server to identify the urgency of a computational request from a handheld. The identification of task priority based upon the location of the mobile device is reflected in the adaptive server response to the computation request. This helps to optimise the quality of service experienced by a handheld device user.

An example deployment scenario of this context-based offloading strategy within a telemedicine environment is shown in Table 1. This table presents an association between the geographic location of a medical practitioner and the priority level assigned to their handheld device computational requests. This priority level reflects the urgency of their offloaded tasks.

Table 1. Priority Levels within Context-Aware Negotiation Protocol

Handheld Device Location	Priority Level
Emergency Room	Urgent
Hospital Ward	High
Hospital Corridor	Medium
Practitioner Office	Low

An examination of related work is presented in section two. In section three, an overview of the context-aware negotiation protocol is outlined. This section details the agent-based architectural framework and the technologies utilised to realise the overall system. Section four depicts an experimental prototype. Finally, section five concludes with an outline of future research.

2 Related Work

There has been a range of research investigating the potential of integrating reconfigurable hardware into the environment of a client system. These research efforts have primarily focused on developing middleware solutions to support client systems in utilising adaptive servers to improve their system performance.

An attempt to establish ubiquitous access to remote reconfigurable hardware has been previously outlined [3]. The objective of this work is to allow a network of reconfigurable hardware modules be transparently accessible by client applications through a JINI-based infrastructure.

Middleware capable of discovering under-utilised computing nodes containing reconfigurable FPGA-based accelerator boards has also been developed [4]. The proposed strategy for sharing remote reconfigurable resources extends an off-the-shelf job management system. This framework enables effective scheduling of client requests for access to remote reconfigurable hardware.

Our previous research identified the potential of using an agent framework to facilitate access to adaptive servers by mobile devices [2, 5]. This initially involved examining the technological feasibility and performance of a handheld equipped with an agent middleware. The introduction of a negotiation and bidding technique for these devices facilitating access to reconfigurable resources within their environment was also investigated. This resource allocation framework employed concepts based upon the contract-net protocol [6]. The negotiation strategy allowed for effective load balancing across all adaptive servers. This maintained a fair workload distribution amongst adaptive servers avoiding both bottlenecks and under-utilisation of resources.

This paper extends our previous research by enhancing the initial negotiation protocol with context-aware capability. This improves the quality of service experienced by a handheld device user by recognising the urgency of their computational requests.

3 Context-Aware Negotiation Protocol

The context-aware negotiation protocol between handheld devices and adaptive servers is embedded into an agent-based architectural framework. This protocol enables a handheld device to efficiently offload reconfigurable hardware-software based computations to neighbouring adaptive servers.

3.1 Agent-Based Framework

The agent-based architecture is shown in Figure 1. This diagram highlights paths of intercommunication amongst agents as well as dynamic agent creation. The architecture was developed using an agent-oriented analysis and design methodology [7]. The number flow within the diagram shows the sequence of activities for a handheld device employing the context-based negotiation protocol. The role of each agent is outlined as follows:

- *Mobile Device Manager*

This single instance agent is a permanent resident on the handheld device and has responsibility for gathering and maintaining information about the physical device and its owner. The agent operates as the main point of contact between the user and the networked reconfigurable resources. The agent responds to resource limitations on the mobile device by attempting to schedule a performance intensive computation

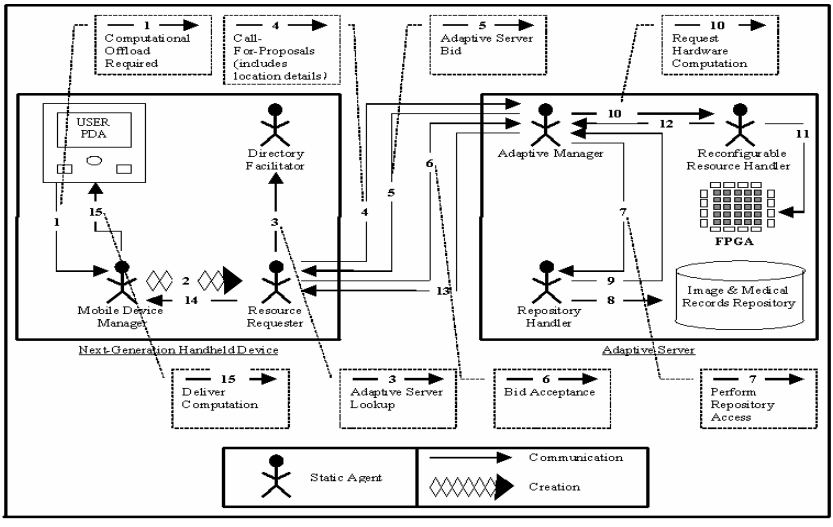


Fig. 1. Agent-Based Architecture

upon the most beneficial of its neighbouring adaptive servers. The mobile device manager initialises this process of selection by creating a resource requester agent.

- *Resource Requester (Negotiator)*

This agent is instantiated as needed and is responsible for initiating the context-aware negotiation process with adaptive servers for access to their reconfigurable resources. This negotiation strategy employs concepts based upon the contract-net protocol. The resource requester agent retrieves a list of all adaptive manager agents within the network from the directory facilitator. A call-for-proposals computation request is broadcast to all adaptive manager agents on this list. The communication request includes details regarding the current location of the handheld device. The resource requester evaluates all adaptive manager bids to determine the best offload option.

The computational task is assigned to the adaptive server which promises to service the request in the quickest time. This process of choosing an adaptive server helps maintain load balancing across all adaptive servers as it ensures fair workload distribution.

- *Adaptive Manager (Negotiator)*

This agent is responsible for facilitating access to reconfigurable resources on an adaptive server. Each adaptive manager submits a bid to execute the mobile device computational request. The bid is determined by examining their current queue of jobs and estimating the total service time. This examination of the queue takes into account the geographic location of the current handheld device request. The priority level associated with the location of the incoming request dictates the placement of the potential computation within the adaptive manager's queue of jobs.

The result of the evaluation combined with an estimate of the time required to service the current computation request determines the adaptive manager bid. The

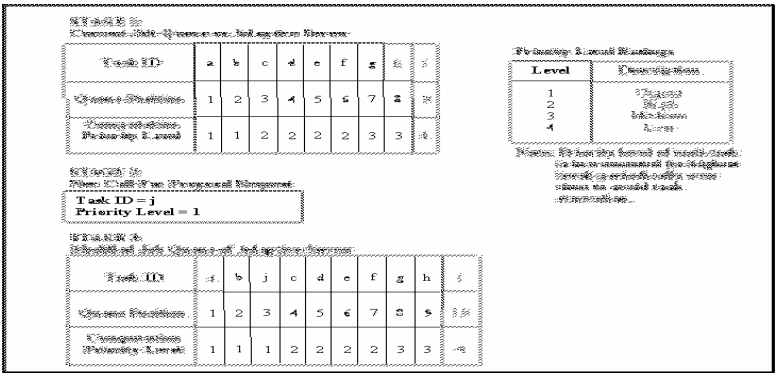


Fig. 2. Adaptive Server Queue Example

location aware aspect of the decision-making process identifies the urgency of the user request. An example scenario of modifications to a queue of jobs by an adaptive manager is shown in Figure 2.

- *Directory Facilitator (DF)*

This agent is responsible for maintaining knowledge about the location and services of each agent within the platform.

- *Repository Handler*

This agent has responsibility for retrieving the reconfigurable bitstream representation of an algorithm required for a handheld device's computational request. The agent is also responsible for returning any additional data that may be required e.g. scanned patient images.

- *Reconfigurable Resource Handler*

This agent is responsible for the process of downloading the bitstream configuration to the reconfigurable resource, interacting with the FPGA and communicating the results of the hardware computation to the adaptive manager.

3.2 System Architecture

The physical framework developed for the context-aware negotiation protocol is presented in Figure 2. The proposed architectural framework utilises JADE (Java Agent Development Environment) as the active agent platform on all provisioning and adaptive servers [8]. JADE is a Java-based open source development framework aimed at developing multi-agent systems and applications. JADE-LEAP (JADE-Lightweight Extensible Agent Platform) is the active agent platform on all mobile devices [9]. JADE-LEAP is an agent based runtime environment that is targeted towards resource constrained mobile devices. Both JADE and JADE-LEAP conform to FIPA (Foundation for Intelligent Physical Agents) standards for intelligent agents. FIPA is a standards organization established to promote the development of agent technology [10].

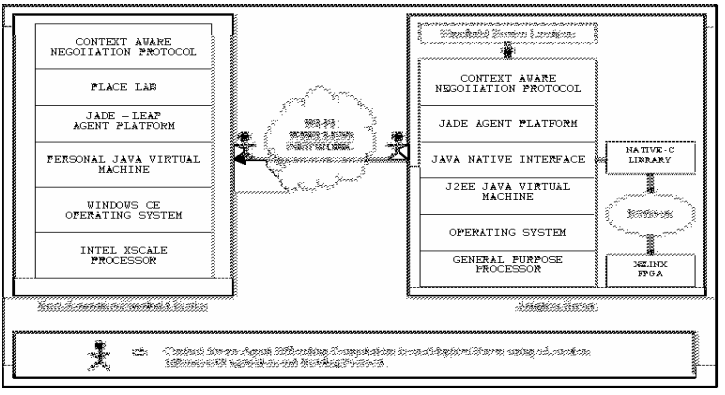


Fig. 3. Physical Framework

3.3 Reconfigurable Technology

The proposed approach to executing a reconfigurable hardware based computation involves an agent interfacing with native C-libraries to dynamically manipulate FPGA circuitry. The native C-Library enables the agent to communicate and configure the hardware portion of its code to the reconfigurable logic. An agent can interact with the FPGA by controlling execution, obtaining feedback, and performing dynamic and partial reconfiguration. The device library provides an interface between an FPGA and the agent platform.

Hardware modules to be configured onto the FPGA are defined in Handel-C. This is a programming language built upon the syntax of conventional C that has additional parallel constructs to gain maximum benefit in performance from the target hardware [11]. Handel-C is used to produce an intermediate hardware format definition of the hardware algorithms (e.g. EDIF, VHDL). These are synthesised to a bitstream configuration for the target FPGA using Xilinx place and route tools [12].

3.4 Location Tool

A contextual element required for successful deployment of the context-based negotiation protocol is knowledge of the location of the handheld device. This is facilitated within the framework through the incorporation of Place Lab technology. This is an open source development project that uses a radio beacon-based approach to location [13].

An agent executing on a portable device can use the Place Lab component to estimate its geographic position. This is achieved by listening for unique identifiers (i.e. MAC addresses) of Wi-Fi routers. These identifiers are then cross-referenced against a cached database of beacon positions to achieve a location estimate.

4 Experimental Prototype

An experimental prototype has been developed to establish the viability of our approach in meeting the future performance and versatility requirements of mobile

devices. This prototype incorporates the context-based negotiation protocol of enabling a portable device offload computation to a neighbouring adaptive server using a location influenced technique.



Fig. 4. Experimental Prototype Screenshots

The prototype system is targeted towards telemedicine imaging applications which are computationally intensive for mobile devices. This prototype enables a physician to retrieve scanned patient images as shown in Figure 3. The left screenshot displays the options available to a physician in terms of their patients, associated scanned images and imaging algorithms that can be applied. The right screenshot shows a patient's original brain scan and a filtered edge-detected image created in real-time by an adaptive server.

Edge detection algorithms are used widely in medical practise to aid physicians in their patient analysis. The performance benefit of implementing an edge detection algorithm with reconfigurable hardware has observed an increase in speed of a factor of twenty in comparison with an implementation of the algorithm in software [14].

The prototype environment consists of a Dell Axim PDA running a Pocket PC 2003 operating system and executing the JADE-LEAP agent platform using a Personal Java implementation of a Java Virtual Machine called Jeode.

The PlaceLab software plug-in resides on the mobile device enabling an accurate location estimate to be communicated to adaptive servers within a call-for-proposals computation request.

Four adaptive servers execute within agent containers on a high-end Pentium PC executing the JADE agent platform. They are connected to a Celoxica RC200 FPGA development board, enabling the execution of a reconfigurable hardware code portion of an offloaded computational request. Agents communicate between the mobile device and the adaptive servers over a Wi-Fi network.

4.1 Performance Results

The effectiveness of the context-based negotiation protocol was evaluated with the development of a purpose built simulator. A Java-Based simulation environment was created and this used data obtained from the prototype system to achieve reliable analysis.

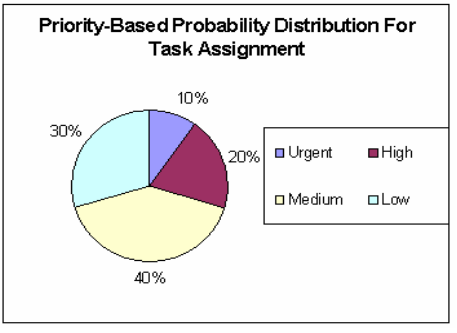


Fig. 5. Priority-Based Distribution Scenario for Task Assignment

A test was conceived to examine the effect on service time for a handheld device offloading a computational task using the context-aware negotiation protocol. The results of this location enhanced approach were contrasted against an agent-based negotiation protocol operating without contextual abilities.

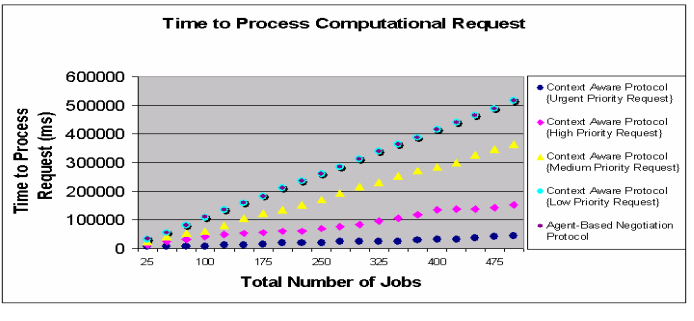


Fig. 6. Time to Process Computational Requests Using Context-Aware Negotiation Protocol

The simulation tool assigned each task a priority level using the Monte Carlo random allocation technique. This task assignment uses the priority-based probability distribution shown in Figure 5. These tasks were then allocated to adaptive server queues equally generating a state analogous to a load-balanced network.

The test case results are presented in Figure 6. These graphs show the time required to service a priority-based computational request from a handheld device using the context-aware negotiation protocol. This service time is dependant on the urgency

of the computational request and the current number of tasks queued for processing by the adaptive servers. The request priority is dictated by the location of the hand-held device. The speed of service delivered by the context-aware protocol is contrasted against the service time of the agent-based negotiation strategy which is non-contextually aware.

A breakdown of the quantitive time values used for simulation testing within the context-aware approach is shown in Table 2. These input values were determined through an averaging of results achieved from twenty separate test-runs of both experimental prototype implementations.

Table 2. Average Quantitive Times for Negotiation Protocols

Implementation Type	Intercommunication Time	Adaptive Server Execution Time
Context-Aware Negotiation Protocol	3320 ms	4081 ms
Non-Context Aware Negotiation Protocol	3110 ms	4075 ms

The results graph shows the time to process a low priority task using the context-based protocol is slightly higher than the time necessary using the non context-aware negotiation strategy. This is primarily due to the slight increase required for computation time within the context-aware approach. The additional execution time is attributed to the added complication of computational requests having associated priorities.

The beneficial effect of the context-aware approach in terms of observing and responding to the urgency of user's computational requests is highlighted within the graph. High priority requests are serviced significantly quicker depending on their priority level. An example of this quicker service can be illustrated when three hundred tasks were distributed and awaiting process amongst the adaptive servers within the simulation environment. A new medium priority task was serviced 31% quicker than a task offloaded using the non-contextually aware negotiation protocol. A high priority task was serviced 73% quicker whilst an urgent priority witnessed 94% faster service.

The primary reason enabling the protocol to deliver this quicker service is its ability to recognise the urgency of each computational task. This facilitates task placement within adaptive server job queues according to associated priority.

5 Conclusions

This paper proposes a context-based negotiation and bidding protocol to enable a handheld device to intelligently utilise surrounding reconfigurable resources. This approach can help next-generation mobile devices improve their system performance and versatility.

The contextual aspect of the protocol uses the location of a mobile device to identify the urgency of a user request. This helps to optimise the quality of service experienced by a handheld user. Our future objectives include integrating learning characteristics into our negotiating agents to enable them to use knowledge of past performance by adaptive servers in their decision-making. This should improve their decision-making ability and enhance their utilisation of networked reconfigurable resources.

Acknowledgements

The work is funded by the Boole Centre for Research in Informatics.

References

1. Compton, K., and Hauck, S.: Reconfigurable Computing: A Survey of Systems and Software. ACM Computing Surveys, vol. 34, no. 2, (2002)
2. O' Sullivan, T., and Studdert, R.: Agent Technology and Reconfigurable Computing for Mobile Devices. In 20th ACM Symposium on Applied Computing, Special Track on Handheld Computing, Santa Fe, New Mexico, USA, (2005)
3. Indrusiak, S., L., et al.: Ubiquitous Access to Reconfigurable Hardware: Application Scenarios and Implementation Issues. In Proceedings of Design, Automation and Test in Europe Conference (DATE), (2003)
4. Gaj, K., at al.: Effective Utilisation and Reconfiguration of Distributed Hardware Resources using Job Management Systems. In Proceedings of the Parallel and Distributed Processing Symposium, (2003)
5. O' Sullivan, T., and Studdert, R.: Handheld Medical Devices Negotiating for Reconfigurable Resources using Agents. In Proceedings of 18th IEEE International Symposium on Computer-Based Medical Systems, Dublin, Ireland, (2005)
6. Smith, R.: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. In IEEE Transactions on Computers, vol. 29, (1980)
7. Wooldridge, M., Jennings, R., N., and Kinny, D.: A Methodology for Agent-Oriented Analysis and Design. In Proceedings of the Third International Conference on Autonomous Agents, 1999.
8. Bellifemine, F., Poggi, A., and Rimassa, G.: JADE – FIPA Compliant Agent Framework. In Proceedings of PAAM, (1999)
9. JADE-LEAP: http://sharon.cslet.it/project/jade
10. Foundation for Intelligent Physical Agents (FIPA): FIPA Agent Management Specification, available at http://www.fipa.org
11. Handel-C Language Reference Manual, version 2.1, Celoxica Limited, (2001)
12. Xilinx Corporation: http://www.xilinx.com
13. LaMarca, A., et al.: Place Lab: Device Positioning Using Radio Beacons in the Wild. In Proceedings of Pervasive 2005, Munich, Germany, (2005)
14. Daggu, V., R., Muthukumar, V.: An Efficient Reconfigurable Architecture and Implementation of Edge Detection Algorithm using Handel-C. In Proceedings of International Conference on Information Technology: Coding and Computing, (2004)

Location-Aware Web Service Architecture Using WLAN Positioning

Ulf Rerrer

Department of Computer Science, Fuerstenallee 11, 33102 Paderborn,
Paderborn University, Germany
urerrer@upb.de

Abstract. The steady rise of mobile computing devices and wireless local-area networks (WLAN) has fostered a growing interest in location-aware systems and services (LBS). Context-awareness in mobile systems give users more convenience with services specific to their preferences, behaviour and physical position. This paper presents an architecture for LBSs in WLANs describing the essential location determination component. An example service using Web Service techniques is described to illustrate efficient service invocation and interaction.

1 Introduction

Recent advances in wireless technology together with demands for an extensive user mobility led to the development and installation of numerous wireless networks and more and more powerful wireless devices. In this world of mobility a multiplicity of new technologies, applications, and services exist. The steady rise of mobile computing devices and local-area wireless networks has fostered a growing interest in location-aware systems and services. Our everyday life changes because Internet applications are nowadays involved everywhere. Activities like gathering information, exchanging documents, e-mail business correspondence, telephone and video conferences, e-learning or web-shopping are done everywhere, anytime.

New opportunities for wireless communication and interaction are provided by so-called location-aware or *location-based systems and services* (LBSs). A key feature of such systems is that the application information and/or interface presented to a user are a function of his or her physical location, a key component of their personal context. The Web offers organisations the possibility to reach potential clients like never before through specialised services. A dynamic matching of providers with requestors can be done with context-sensitive information. Extensive work concerning Web Services (WSs) has improved the way to build applications with dynamic service discovery and efficient utilisation of Web resources.

The goal of this work is to describe an architecture for location-based services using a wireless LAN (WLAN) positioning technique. Gathering automatically location information is used to easily develop convenient context-aware services combining advantages of Web Service technology and indoor location determination. This paper introduces a file-sharing example service for meetings where

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 196–205, 2005.
© Springer-Verlag Berlin Heidelberg 2005

participants – adapted to their local environment – can access and provide documents for other members without any pre-configuration or security legitimation.

The remainder of this paper is organised as follows: In section 2, we survey related work in all relevant research fields concerning WSs, location determination in wireless networks and LBSs. In section 3 the scenario of LBSs in WLANs is described in detail. Section 4 presents a closer look at the proposed LBS system including its architecture, used positioning technique, and realised services. Conclusions are future activities are described in the final section.

2 Related Work

Location-based services in WLANs is a new research field and our approach requires techniques from different environments. Related work can only be found for each scope (positioning, LBSs, WSs) separately.

2.1 Positioning in WLAN

To provide location-based services a technique to determine the position of a mobile device in a wireless network is necessary. LBSs are already established in mobile phone networks. Triangulating a mobile phone's signal from several base stations using precise clocks and other hardware allows a location determination with a relatively high accuracy. Satellite networks such as the Global Positioning System (GPS) require special hardware. GPS has problems with signal shadowing in cities and only work properly outside of buildings [10].

Transferring LBSs to an indoor wireless network, especially WLAN, requires an accurate positioning system and special attention in the system development. WLAN devices like notebooks or PDAs are much more capable than mobile phones, but usually not locate-able precisely. The most promising technique dealing with radio frequency (RF) signals is to process the received signal strength (RSS) of a mobile device. One of the first positioning systems in WLAN was RADAR [11] building on conventional WiFi hardware unlike other approaches using infrared [12], ultrasound [13] or RFID-Tags [14]. Special characteristics of WLAN make this technology ideal for location-based services. The broad availability and acceptance of WLAN reached in recent time combined with the mobility, affordability and flexibility of the used devices offer an effective and low-cost solution for LBSs.

2.2 Location-Based Services

The emerging world of mobility is characterised by a multiplicity of exciting new technologies, applications, and services. Amongst the most promising ones will be the ability to identify the exact geographical location of a mobile user at any time. This ability opens the door to a new world of innovative services, which are commonly referred to as location-based services. In traditional positioning systems, location information has typically been derived by a device

(GPS receiver). However, widespread interest in LBSs boost in late 1990s in mobile phone networks and comes to any kind of wireless network today. It presents many challenges in terms of research and industrial concerns [15,16]. User location is an important dimension in this new data-service world. A more detailed description of LBS and its convenience is presented in the next section.

2.3 Web Services

Web Services go beyond conventional middleware for true application integration. From numerous definitions the one fits most with our intended use of WSs is given in [1] where WSs as defined as "loosely coupled, reusable software components that semantically encapsulate discrete functionality and are distributed and programmatically accessible over standard internet protocols". This definition captures the self-contained, modular, compose-able and distributed nature of WSs.

The benefit of Web Services relies on the ability to publish, find, browse, select, compose, employ and monitor services in the Web. WSDL [2] allows definition of a service interface and service implementation. The SOAP [3] protocol offers basic communication via message passing. XLANG [4], WSFL [5] and BPEL4WS [6] support process modelling at a syntactic level to create complex services. Acceptance in industry accomplished a wide variety of WS implementations gathering experiences for commercial use [7].

Often human interaction is necessary to use Web Services properly. As an example UDDI [8] provides an XML-based schema for describing Web Services, but to determine if the service is what you need, it is necessary to contact the provider, or to browse several documents referred to in the UDDI specification. Future research considers agents who automatically find, compose and execute services which are in the line of action of the Semantic Web Initiative [9].

3 Location-Based Services

To get an impression on the potential of LBSs this section provides a closer look to common scenarios, some service classification, and the basic workflow for our approach describing the convenience of a service-oriented architecture providing location-aware Web Services.

3.1 Classification

The term *location-based services* is a concept that denotes applications integrating geographic locations with the general notion of services. Many examples already exists including applications for emergency services, car navigation systems, "yellow maps" as a combination of yellow pages and maps or location-based billing systems. These applications can be classified in:

- Navigation aid (for people and robots)
- Shopping assistance (customer behaviour tracking)

– Information Services (advertisement of restaurants, movies etc.)
– Tracking of assets (assets in hospitals, factories, military etc.)

A major distinction of services is whether location-based services applications are *person-oriented* or *device-oriented*. In addition to this the application design can be distinguished in *push* and *pull services* [17].

3.2 Application Areas

Different application areas exist in the world of location-aware systems and services. Many of them are based on different technologies or target different audiences. The granularity of position information needed could vary from one application to another. For example, locating a nearby printer requires coarse-grained positioning information whereas locating a book in a library requires more fine-grained information.

An intuitively setting for LBSs are *natural disasters*. Disasters like fire, floods, earthquakes, volcano eruptions, hurricanes, etc. or technical/material errors can cause severe damages to buildings, properties and even humans affecting large numbers of people. The coordination of professional and rapid help provided by aid agencies, fire fighters, police, etc. requires a significant amount of management infrastructure and keeps many people busy during a crisis. LBSs can help in this difficult situations based on new technologies such as GPS, sensor networks and wireless LANs assessing damages, identifying tasks, and coordinate different aid organisations.

Standard *tourist information systems* also benefit from today's small and high-performance wireless devices in combination with LBSs. High-precision positioning techniques deliver accurate location information to develop tourist guides with detailed background information to exhibits even outside museums [18]. Finding the nearest restaurant, hotel or other points of interest as well as finding close-by friends are well known and already established services in mobile phone networks [19].

Another interesting class of location-aware services is an *office scenario*. Printing to the nearest printer, navigating through a building to the next meeting, finding a colleague on a large floor, etc. adds value to standard wireless data networks [11]. Filtering vast information sources well adapted to the users location often useful if presented automatically.

3.3 Scenario

The basic workflow for each area mentioned above is similar. The office scenario could be described as follows: A user with a mobile device enters an office building. The building is equipped with a wireless LAN infrastructure for communication and a LBSs system. When the user moves within the range of one or more WLAN access points his/her wireless device is recognised by the system. Standard WLAN connectivity procedures care for authentication, IP addresses distribution via DHCP, and many more matters when devices log into the network. A *splash screen* technique – well known from airport lounges or other

commercial WiFi hot spots – shows a welcome screen as first homepage in a browser. Instead from forcing a user to enter his credit card number to surf the Web an introducing splash screen for the location-based system is shown. Here the user is advertised about the LBSs in the surrounding area. When the user has chosen one or more services this selection is stored in the LBS system and a simple interaction via browser takes place like using other standard WSs.

For example the user wants to go to the conference room of his imminent meeting. Therefore, he uses the *map service* entering his target room and a map of the building is shown with the shortest route from his current position to the conference room. On the way the user wants to print the latest version of his slides for his presentation in the meeting. He uses the *printing service* which determines the nearest available printer matching the requirements to print coloured slides. The map service updates the map and an alternative route including a stopover at the printer is now shown.

For the meeting a *file-sharing* service is activated. During the meeting all participants in the conference room – and no one else from outside – can easily share their relevant documents such as slides, reviews of videos via a public folder. This is achieved by a file-sharing service providing all users in a specific room a common web space for the meeting. This group is defined by the location of the users only. No pre-configuration or registration is needed to grant admission to the restricted files.

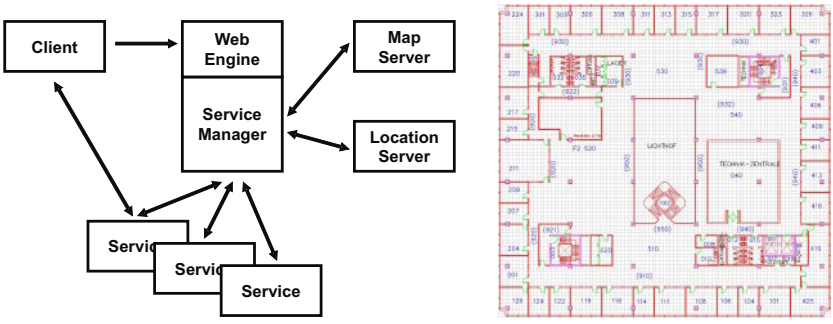
Shortly before leaving the meeting a *transportation service* takes action. Knowing the next appointments from the users calendar the service revises the location of the next meetings scheduled. The next meeting is in a different building at the other side of the city. The transportation service determines the relevant bus stations of the two buildings due to the current location of the user in the LBS system and the location specified for the other building in the calendar. After that the shortest route by bus is determined and the departure time together with a map to the bus station in front of the building is shown on a map from the service. After the meeting the user leaves the conference room, follows the route to the bus station and drives to the second location having another meeting.

This scenario denotes the power and comfort of location-based services. A central *management component* – the LBSs system mentioned in the paragraph above – is needed to register the users of such a system and coordinate the communication between user and services. It is essential to determine the exact position of the user in the building. A *location component* determines the position of a wireless device via triangulation of the received signal strength from different access points. A more detailed description of the architecture is given in the next section.

4 The LBS System

4.1 Architecture

The main component of the location-based service system is the service-manager including a web-engine for service presentation and interaction (figure 1(a)). A



(a) LBSs system architecture (b) Building map of top floor

Fig. 1. System architecture and environment map

small client application is necessary for the positioning which is processed in the location-server component. The map-server is another important part of the architecture generating maps from the coordinates given from the location-server. The encapsulated services mainly communicate with the service-manager via XML interfaces.

The *service-manager* implements the central interface to the system and dispatches all main events from user registration, over service management, to communication bridging. This component also includes a *web-engine* for different purposes. First, it contains a NoCat [20] server to realise the splash-screen technique noted in section 3.3 advertising the entire system and acting as gateway to the LBSs system. It forwards the communication between service and client using ISL [21] (Interaction Specification Language based on XML) generating a device-dependent presentation (notebook, PDA, mobile phone, etc.) with the help of XSL-stylesheets.

The *location-server* implements the core for the position estimation and holds a database with reference values from received signal strength measurements. The component receives a vector from the client with current values and performs several algorithms [11,22,23,24] to determine the clients' position. As result coordinates are given to the service-manager for further processing.

The *map-server* for example is a basic service implementing a map service generating dynamically multi-layered vector-maps with miscellaneous objects and their positions. Rooms, floors, devices with dynamic properties and users can be displayed and even routing between objects can be done.

The other *services* have an interface for basic communication and interaction with the main component. Other interfaces create service specific connections directly to the client. With the help of these interfaces services like the above mentioned transportation service or file-sharing service are possible.

The *client* itself is usually a mobile device equipped with a wireless LAN unit. Due to the broad variety of devices with most differential display capabilities only very weak requirements for the client hard- and software are made. The LBS system only needs a browser which is almost always present in this class

of devices. Therefore, any service can display and interact with the user via the visual capabilities of a standard browser.

4.2 Location Server

The *location server* is an essential part of the purposed LBS system. The experimental environment is the computer science building of our university with four floors each 56m x 52m large. Figure 1(b) shows a map of the top floor. On each floor are at least four access points (AP) containing standard IEEE802.11b WiFi network cards, equipped with external omnidirectional antennas. The mobile equipment for the testbed environment includes a wide variety of notebooks (eight different vendors), network cards (four different vendors of PCMCIA cards) and operating systems (Windows XP, Linux, Windows Mobile 4.2x) to serve a broad diversity of users.

The location server holds a set of reference vectors building a radio map of each floor. An almost regular grid – due to the presence of various walls and obstacles – with 2.4m spacing on the map defined the points for the reference measurements. Therefore, on more than 500 points each floor the received signal strength of all reachable access points was measured and stored in the location server's reference database. Radio maps of two access points on the same floor are shown in figure 2.

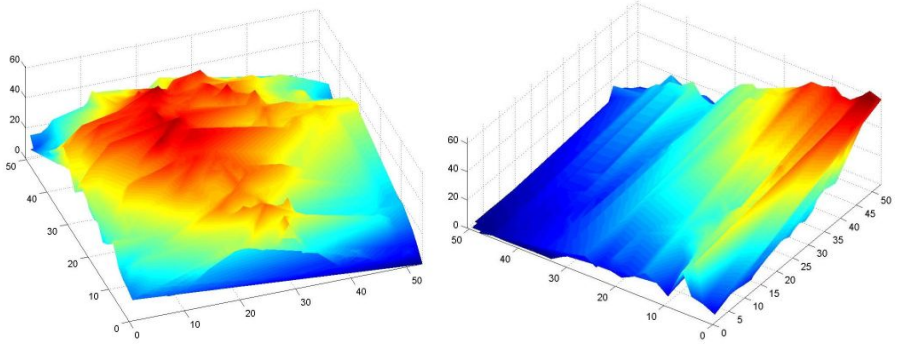


Fig. 2. WLAN signal strength of two APs on the same floor

The implemented algorithm for the location determination is a sort of *nearest neighbour in signal space*. At first we used in our algorithm the *Euclidean distance* (Equation 1) between each reference vector $\boldsymbol{r} = (r_1, \ldots, r_j) \in \boldsymbol{R}$ and the recorded signal strength at the location to be determined $\boldsymbol{x} = (x_1, \ldots, x_j)$ where j is the number of APs and $\boldsymbol{R}$ is the set of reference vectors.

$$\min_{r \in \boldsymbol{R}} ||x - r||_p := \left(\sum_{i=1}^{n} |x_i - r_i|^p \right)^{\frac{1}{p}} \qquad \text{where } p = 2 \tag{1}$$

Instead to pick the location that minimises the distance we aggregated *multiple nearest neighbours*. We took the best k reference values (smallest difference to $\boldsymbol{x}$: $m_1, m_2, \ldots, m_k$, see Equation 2) and determined the *intersection point* as estimated location. Our analysis shows that for small k, averaging the position has some benefit on the result. If k becomes greater than 5 this effects has no significance any more.

Further we experienced that the signal strength values in the database only had small differences. Therefore, we changed the value of the exponents in the above equation to four to amplify the differences and not the absolute values. This rather simple algorithm achieved great performance in the location server. The accuracy of the location estimation is usually below 2 meters.

$$\begin{aligned} m_1 &= \min_{r \in \boldsymbol{R}} ||x - r||_4 \ , & m_1 \in \boldsymbol{R} \\ m_2 &= \min_{r \in \boldsymbol{R} - \boldsymbol{m_1}} ||x - r||_4 \ , \ldots & m_2 \in \boldsymbol{R} \\ m_k &= \min_{r \in \boldsymbol{R} - \boldsymbol{m_1}, \ldots, \boldsymbol{m_{k-1}}} ||x - r||_4 & m_k \in \boldsymbol{R} \end{aligned} \tag{2}$$

Future work on this component includes tracking and probabilistic algorithms to reduce "jumps" of users/devices in a short period of time. This happens when nearest neighbours in signal space have greater distances in the real world.

4.3 File-Sharing Service

The idea of the file-sharing service was briefly mentioned in section 3.3. This information service realises a simple device-oriented example to show the comfort and usage of the system architecture. In a meeting, for example, a quick and easy document sharing among the participants is often necessary. The file-sharing service provides access to a restricted web space where any kinds of documents can be uploaded and downloaded without any special configuration. This pulling service gains access control by the usual login and the specific location of the meeting members. Together with the users' preferences and usual habits a powerful context-aware service can be created.

The service is implemented as Web Service. The service provider registers the services URI at the LBS system and generates the WSDL description of the service for interaction. The obviously advantage of this solution is the easy access through the Internet without any conventional middleware such as CORBA or RMI in other distributed architectures and that interaction between the user and the system can be done through a standard web browser. A more hidden advantage arises in service-to-service communication. Automatic service description and discovery are realisable with static or dynamic binding resulting in service with service interaction and composing complex services from a set of basic services.

The access control of the file-sharing service interacts with the location server. The meeting coordinator defines the *outer parameters* like time and room for the meeting. The service itself derives the *inner parameter* from these. Two zones (see figure 3) are defined in a rectangular or polygonal shape covering the entire

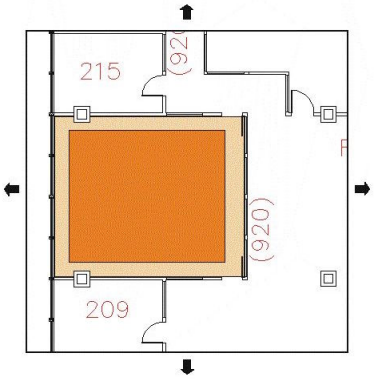


Fig. 3. Map with file-service zones

room. In this *notification zone* each user who accesses the service is asked to go a little further in to the centre of the room. Within this zone a smaller *access zone* is defined. In this zone a user can upload and download documents to a prepared web space. The expansion of the inner zone is regulated automatically by the LBS system's positioning accuracy eliminating secret access from users outside the room. With established access participants of the meeting can easily share documents just by a few clicks in their web browser. No configuration or setups of FTP-servers, user-groups or web-space access is needed. This example shows the convenience of location-based services.

5 Conclusions and Future Work

Location-based services in wireless LANs are a new and emerging research topic. Context-aware mobile systems need a sophisticated architecture and an effective communication and cooperation of the components. This paper presents such a service-oriented architecture describing a location determination technique that delivers the physical location of a user in an indoor wireless network. The exemplarily described file-sharing service realises a context and location-aware service using Web Service techniques for service invocation and interaction.

Future work includes improving the accuracy of the location server with device tracking and probabilistic location determination. Semantic service descriptions like in the Semantic Web Initiative to enhance the automatic discovery, composition and execution of services, are also considered.

References

1. Stencil Group "Defining Web Services" `http://www.stencilgroup.com`, 2001.
2. E. Christensen, F. Curbera et al. "The Web Service Description Language WSDL" `http://www-4.ibm.com/software/solutions/webservices/`, 2001.
3. W3C "SOAP Specification" `http://www.w3.org/TR/soap/`, 2000.

4. S. Thatte "XLANG, Web Services for Business Process Design" http://www.gotdotnet.com/team/xml-wsspecs/xlang-c/, 2001.
5. F. Leymann "WSFL, Web Services Flow Language" http://www-4.ibm.com/software/solutions/webservices/, IBM, 2001.
6. F. Curbera et al. "Business Process Execution Language for Web Services" http://www-106.ibm.com/developerworks/webservices/library/wsbpel/, 2002.
7. W3C "Web Services Community" http://www.webservices.org, 2005.
8. UDDI "Universal Description, Discovery and Integration for Web Services" http://www.uddi.org/pubs/, 2001.
9. W3C "Semantic Web Community" http://www.semanticweb.org, 2005.
10. S. H. Cobbs "GPS Pseudolites: Theory, Design, and Applications" Ph. D. thesis, Stanford University, USA, 1997.
11. P. Bahl and V. N. Padmanabhan. "RADAR: An In-Building RF-based User Location and Tracking System" Proceedings of IEEE InfoCom, pp. 775–784, 2000
12. R. Want, A. Hopper, V. Falcao and J. Gibbons "The Active Badge Location System" ACM Transactions on Information Systems, pp.91-102, Vol.10, No.1, 1992
13. N. Priyantha, A. Chakraborty and H. Balakrishnan "The Cricket Location Support System" Proceedings of the ACM/IEEE Mobicom, pp.32-43, 2000
14. L. M. Ni, Y. Liu, Y. C. Lau and A. P. Patil "LANDMARC: Indoor Location Sensing Using Active RFID" Proceedings of the IEEE PerCom, pp.407-415, 2003.
15. S. J. Barnes "Location-based Services: State of the Art" e-Service Journal, 2003
16. J. Hightower and G. Borriello. "Location Systems for Ubiquitous Computing" IEEE Computer Journal, 34(8), pp.57–66, August 2001
17. J. Schiller and A. Voisard "Location-based Services" Morgan Kaufmann, 2004
18. R. Kramer, M. Modsching and K. ten Hagen "Context based Navigation by a Dynamic Tour Guide" Workshop on Positioning, Navigation and Communication (WPNC), 2005
19. T. D'Roza, G. Bilchev "An Overview of Location-based Services" BT Technology Journal, vol. 21 no.1, Jan. 2003
20. NoCat "NoCat Server" http://www.nocat.net, 2005
21. S. Nylander and M. Bylund "Device Independent Services" SICS Technical Report T2002-02, Swedish Institute of Computer Science, 2002.
22. P. Castro, P. Chiu, T. Kremenek, and R. Munz "A Probabilistic Room Location Service for Wireless Network Environments" Proceedings of IEEE Ubicomp, 2001
23. M. A. Youssef, A. Agrawala, and A. U. Shankar "WLAN Location Determination via Clustering and Probability Distributions" Proceedings of IEEE PerCom, 2003
24. U. Rerrer, O. Kao "Suitability of Positioning Techniques for Location-based Services in wireless LANs" Workshop on Positioning, Navigation and Communication (WPNC), pp. 51-56, 2005.

A Light-Weight Framework for Location-Based Services

W. Schwinger, Ch. Grün, B. Pröll, and W. Retschitzegger

Johannes Kepler Universität Linz, Altenbergerstrasse 69, A-4040 Linz
http://www.uni-linz.ac.at

1 Motivation

Context-aware mobile systems aim at delivering information and services tailored to the current user's situation [1], [10]. One major application area of these systems is the tourism domain, assisting tourists especially during their vacation through location-based services (LBS) [4], [7]. Consequently a proliferation of approaches [2], [5], [8], [9], [12], [15], [17], [18] can be observed, whereby an in-depth study of related work has shown that some of these existing mobile tourism information systems exhibit few limitations [3], [19]: First, existing approaches often use proprietary interfaces to other systems (e.g. a Geographic Information System – GIS), and employ their own data repositories, thus falling short in portability and having to deal with time consuming content maintenance. Second, often thick clients are used that may lack out-of-the-box-usage. Third, existing solutions are sometimes inflexible concerning configuration capabilities of the system. To deal with those deficiencies, we present a lightweight framework for LBS that can be used for various application domains. This framework builds on existing GIS standards, incorporates already available Web content, can be employed out-of-the-box, and is configurable by using a Web-based interface. The applicability of the framework is demonstrated by means of a prototype of a mobile tourist guide.

2 Basic Features of Our Framework

In the following a brief overview of the basic features of our framework is given.

Support for LBS. Our framework supports the creation of LBS tailored to the user's position and preferences for arbitrary application domains, e.g. tourism or infrastructure management. The user is provided with a map of his/her surroundings together with points of interests (POIs) and the current position based on GPS.

Integration of external data-sources. Integration of existing external data sources is enabled by incorporating existing GIS servers as well as by augmenting the POIs with existing Web content (cf. e.g. [11], [14]).

Exploitation of GIS standards. Our framework uses the open OGC Web Map Service (WMS) standard [13] for retrieving geospatial information in form of maps.

Application of a thin client approach. A thin client approach is employed allowing to run the application out-of-the-box. On the client side only a graphical, ActiveX enabled browser, an Internet connection and a GPS sensor is required.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 206 – 210, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Configuration of external content inclusion. Our framework offers the possibility of configuring the inclusion of external Web content through a Web-based interface. First, POIs can be added, deleted and updated in the framework's repository (cf. Section 4). Second, for each POI a title, the geographical coordinates and an URL pointing to external Web content describing the POI, can be configured. Third, POIs can be assigned to possibly nested categories, thus forming thematic layers such as gastronomy including the sub categories of cafes, fast food, and bars. Forth, for each category the visualization of the corresponding POIs can be chosen.

Configuration of external GIS inclusion. In addition to configuring the inclusion of the external content in terms of POIs, the WMS request to the external GIS server can be configured again through a Web-based interface. This comprises the URL of the GIS server, the type of map (e.g. aerial photography), which area to retrieve (in terms of coordinates), the spatial reference system (e.g. Gauss-Krüger), the desired output format (i.e. JPEG, GIF, PNG or SVG) and the output size of the map in pixels.

3 Functionality of Our Framework

The main functionality provided for the user is described in the following. A screen shot of the system's graphical user interface is depicted in Fig. 1.

Basic map functionality. Basic map functionality like panning, selecting or changing the zoom level are supported and the user can switch between a street map and an aerial photo.

Selection of thematic layers. The user can select different thematic layers he/she is interested in. The corresponding set of POIs including those POIs associated with sub-categories is then superimposed on the map through pictograms at each user request, thus enabling the user to build a topographical mental model of his/her surrounding (cf. Fig. 2a).

Information about POIs. POI pictograms indicate whether by clicking on them further information, e.g. historic information or pictures can be requested (cf. Fig. 2b).



Fig 1. User Functionality

Search of POIs. The user is able to search POIs by indicating a thematic layer and a certain distance as search radius. Based on this information and the current user's position, POIs are filtered and displayed accordingly.

Refresh of maps. The displayed map is refreshed automatically either as soon as the user performs an explicit action, e.g. panning the map or periodically to reflect the movement of the user. The refreshing period can be adjusted to the user's speed.

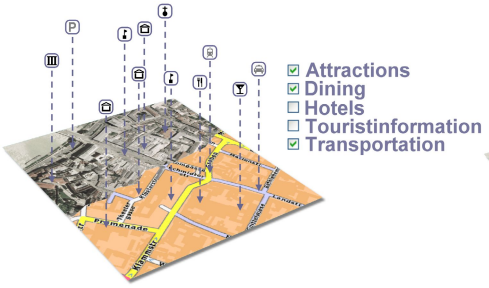


Fig 2a. Thematic Layers

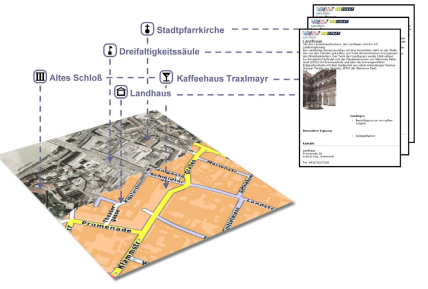


Fig 2b. Inclusion of External Web

Visualization of current position. Getting a valid GPS signal in the historical parts of a city is a commonly known problem, due to the alleyways, which block signals of satellites located near the horizon. Unavailability and changing accuracy is dealt with in the following way: First, we use a circle as the symbol to represent the uncertainty of the user's current position. Second, the user is continuously informed about the quality of the received GPS signal. Third, in the absence of a GPS signal, the centre of the currently displayed map is assumed to be the user's current position.

Collaboration functionality. Since social factors are important in several application domains of LBS, some basic collaboration functionality is provided. First, POIs provide a link to a guestbook allowing users to annotate POIs (e.g. in the tourism domain rating the quality of a restaurant), that can be shared with other users. Second, the system offers the visualization of other users simultaneously using the system, thus providing the prerequisite for subsequent interaction.

4 Architecture of Our Framework

The client/server architecture of our framework consists of internal components in terms of a *LBS Engine* and a *Repository* (cf. Fig. 3). The LBS engine is implemented using JSP and Java Beans and realizes the framework's core functionality. The repository utilizes a database for storing configuration information. External components comprise a *GIS Server* and external *Web Servers* to deliver further information about POIs. On each request the system performs the following six steps:

Step 1: The Web Browser communicates with the Web Server over WLAN and is continuously informed about the user's location through an ActiveX component included in the corresponding HTML-page. As the user invokes actions, e.g.

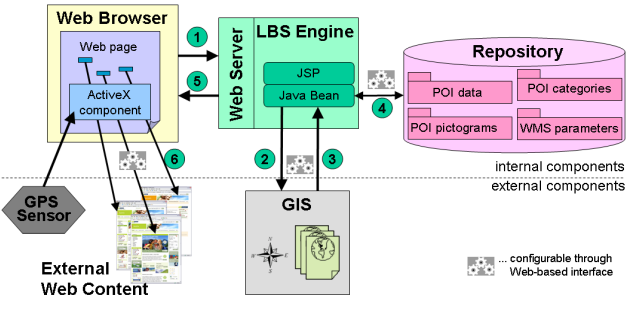


Fig. 3. Architecture

clicking on a POI, a request with the GPS data as additional parameter is sent to the Web Server and forwarded to the LBS Engine processing.

Step 2 and 3: Corresponding to the user's position and kind of request, the LBS Engine retrieves from the Repository the WMS parameters and requests geo-data from the GIS Server using the OGC WMS specific *Get Map Request*. The GIS Server processes the request and returns an image based on those parameters.

Step 4: Before being deployed to the Web Browser, the map image is, based on the user's preferences and location, augmented with information about POIs retrieved from the Repository. The POIs' geographical coordinates are transformed into pictures' pixel coordinates to correctly locate the POI pictograms on-top of the map.

Step 5: The user receives the image in form of a MIME-type encoded picture, which contains the map and thematic layers.

Step 6: When the user requests further information of a specific POI, the corresponding Web page is retrieved from the appropriate external Web Server.

5 Prototypical Application – Linzer Mobile Guide (LiMoG)

The applicability of our framework is demonstrated by realizing a tourist guide for the city of Linz called LiMoG - short for Linzer Mobile Guide. The configured POIs include, e.g., historic sights, churches and cafes along traditional sightseeing tours through the historic centre of Linz. DORIS [6], the GIS of the federal state of Upper Austria, provides the geospatial data in form of an OGC WMS. Each POI links to a certain existing Web page provided by Austria's official destination information and booking system TIScover[16], [20]. For example in case of a restaurant, TIScover provides information about facilities, cuisine, opening hours, prices as well as photos. For demonstration purposes, we used a Pocket PC (IPAQ 5450) equipped with a GPS receiver (SysOn GPS CF plus II). For accessing the Internet, a publicly available WLAN hotspot in the city centre of Linz was utilized.

6 Outlook

Giving the experience gained we intend to extend the notion of context beyond location and user preferences to consider also other context properties (e.g. time, network) as well as a combination thereof. To increase flexibility of the content incorporated, an adaptation component is envisioned, which can transform the content before being displayed to the user (cf. [10]).

References

[1] Altmann, J. et al.: Context-Awareness on Mobile Devices - the Hydrogen Approach, Proc. of the 36th Hawaii Int. Conf. on System Sciences, Hawaii, 2003

[2] Anegg, H. et al.: Designing a Location Based UMTS Application, Springer, 2002

[3] Baus, J. et al.: A Survey of Map-based Mobile Guides. In Liqiu Meng and Alexander Zipf (eds.): "Map-based mobile services – Theories, Methods and Implementations". Springer, 2005
[4] Berger, S., Lehmann, H., Lehner, F.: Location-based Services in the tourist industry, International Journal on Information Technology & Tourism, Cognizant, 5/4, 2003
[5] Cheverst, K. et al.: Developing a context-aware electronic tourist guide: some issues and experiences, Proc. of the SIGCHI Conference on Human Factors in Comp. Systems, The Netherlands, 2002
[6] DORIS, Digitales Oberösterreichisches Raum-Informations-System, http://doris.ooe.gv.at.
[7] Garzotto, F. et al.: Ubiquitous Access to Cultural Tourism Portals**,** Proc. of the Int. Workshop on Presenting and Exploring Heritage on the Web (PEH), Spain, 2004
[8] Hinze, A., Voisard, A.: Location- and Time-based Information Delivery in Tourism, Proc. of the 8th Symposium on spatio-temporal databases, Santorini Island, Greece, 2003
[9] Kamar, A.: Mobile Tourist Guide (m-ToGuide). Deliverable 1.4, Final Report, 2003
[10] Kappel, G. et al.: Customisation for Ubiquitous Web Applications - A Comparison of Approaches. Int. Journal of Web Engineering and Technology, Inderscience Publishers, 2003
[11] Kapsammer, E. et al.: Bridging Relational Databases to Context-Aware Services, Proc. of the CAiSE Workshop Ubiquitous Mobile Information and Collaboration Systems, Portugal, 2005
[12] Krösche, J. et al.: MobiDENK-Mobile Multimedia in Monument Conservation. IEEE MultiMedia, 11/2, 2004
[13] OGC. Open Geospatial Consortium. http://www.opengeospatial.org.
[14] Pashtan, A. et al.: Personal Service Areas for Mobile Web Applications. IEEE Internet Comp. 8/6, 2004
[15] Poslad, S. et al.: CRUMPET: Creation of User-Friendly Mobile Services Personalised for Tourism. In: 2nd Int. Conf. on 3G Mobile Communication Technologies, UK, 2001
[16] Pröll, B., Retschitzegger, W.: Discovering Next-Generation Tourism Information Systems - A Tour on TIScover, Journal of Travel Research, Sage Publications, Inc., Vol. 39/2, 2000
[17] Roth J.: Context-Aware Web Applications Using the PinPoint. IADIS International Conference WWW/Internet, Portugal, IADIS Press, 2002
[18] van Setten, M., Pokraev, S., Koolwaaij J.: Context-Aware Recommendations in the Mobile Tourist Application COMPASS. Proc. of 3rd Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems, The Netherlands, Springer, 2004
[19] Schwinger, W. et al.: Context-Awareness in Mobile Tourism Guides – A Comprehensive Survey. Technical Report, Johannes Kepler University Linz, IFS/TK, 2005
[20] TIScover Destination Management System, http://www.tiscover.com.

Context Awareness for Music Information Retrieval Using JXTA Technology

Hyosook Jung and Seongbin Park*

Department of Computer Science Education, Korea University,
Anam-dong, Sungbook-gu, Seoul, Korea
{est0718, psb}@comedu.korea.ac.kr

Abstract. The development of mobile devices and wireless networks made it possible for users to seamlessly use different devices to recognize changes in their computing environment. In this paper, we propose a music information retrieval system (MIRS) that exploits various types of contextual information. Our system is based on JXTA technology which enables to join at any time and direct communication between peers. It supports the personalized retrieval of music information at specific moments and locations. Each peer can run a context interpreter and each edge peer functions as a query requester or a query responser. The query requester's context interpreter analyzes user's context and customizes the search result. The query responser's context interpreter analyzes the pattern for the query melody and selects requested music information.

1 Introduction

The rapid development of mobile technology allows people to use information with mobile devices such as PDAs and mobile phones anywhere anytime. In this new computing environment, context-awareness is one of the most important issues for providing relevant information[1]. Information retrieval (IR) systems on mobile devices must consider more complex contextual data than desk-based systems because the context can affect what information is relevant to users[3]. Especially, the new working environment demands on a new network environment which enables networked devices to provide or use services between peers effectively as the performance of mobile devices increases[2].

In this paper, we consider the problem of content-based music information retrieval (CBMIR) and propose a context interpreter that processes various types of contexts in CBMIR in a JXTA network that is an ad hoc, multi-hop, and adaptive network composed of connected peers[4]. Music information retrieval in a peer-to-peer environment has been addressed in [6,7,8] and searching music in this environment necessitates to consider various types of contexts since depending on the context, each user may construct a query melody differently. More specifically, we consider three types of contexts in this paper. First, the user context refers to two types of features that are relevant to a user. One is called

* To whom correspondence should be addressed.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 211–214, 2005.
© Springer-Verlag Berlin Heidelberg 2005

physical context and describes features of mobile devices such as type, screen size, capability, etc. a user is using. The other is called personal context and describes user's domain knowledge level, system knowledge level, educational experience, purpose of retrieval, etc. Second, the domain context describes features of the musical content such as a key of music, pitches of each note, pitch name sequences of a melody, etc. Third, the pattern context describes features of a given query melody such as a main chord of the query melody and indices of the most frequent nonharmonic note.

2 Context Interpreter

The context interpreter is responsible for recognizing contextual information. In figure 1, user's context interpreter analyzes the user context and creates the personalized input and output interface. MIRS's context interpreter analyzes pattern context of a query melody, determines a retrieval method, and selects requested information of music with matching melody.

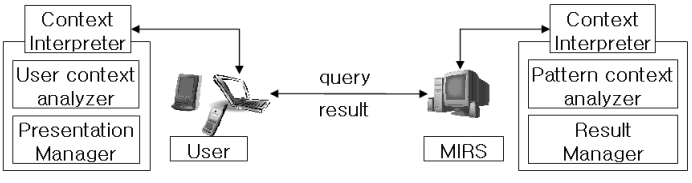


Fig. 1. The context interpreter consists of four modules which perform different functions

The user context analyzer parses a user profile and decides the input items to be included in the input interface. For example, if a user has the high-level domain knowledge and experience, it includes detailed music key list and input form to divide melodies per phrase. It selects different style sheets according to the user's device such as PDA, mobile phone, or notebook. It also decides the arrangement of the input items.

The presentation manager extracts the content items of retrieved result such as composer, title, musical note, etc as well as the values for personal and physical contexts and decides the output items. For example, musical note is shown at notebook while eliminated at cell phone because of screen size. It also selects proper style sheets according to device type and determines the arrangement of the output items.

The pattern context analyzer converts the MIDI numbers of the query melody to pitch name sequence and finds the key information. It determines a main chord of the query melody such as tonic, dominant or subdominant chord by checking chords of each note. It finds all nonharmonic notes centering the main chord and stores indices of the most frequent nonharmonic note. The indices are used to search all patterns which are similar to the query melody. If the matching

melody is not found, it transposes pitch name sequences which are shifted up or down pitch of the query melody by a constant interval because it is not easy for users to memorize the accurate key of melody. Then it analyzes the pattern context again.

The result manager finds user's retrieval goal and optional items with content of the query melody such as additional music description, MIDI file, musical image file,etc. It selects relevant and requested information of music which includes matching melody.

3 Architecture of the System

Figure 2 depicts the architecture of the proposed system. Each peer is assigned a role according to its task goal such as client, rendezvous, and music peer.

When a client peer initiates a discovery request for music peers, it is sent to a rendezvous peer that maintains a cache of advertisements and forwards discovery requests to help client peers to discover resources. The client peer contains a user profile which records personal and physical features such as domain knowledge, interest, education, job, type of device, screen size and so on. The client peer attempts to connect using the information delivered from the rendezvous peer. After executing context interpreter, the client peer provides the personalized input interface where a user enters a query melody and sends it to the music peer. After executing context interpreter, the music peer receives the query melody and searches matching melody based on the pattern and domain context of the query melody which are found by pattern context analyzer. The retrieval engine presearches all melody patterns which are similar to the pattern context of the query melody in music database, using a vowel-based algorithm[5]. Then, it recognizes music that includes the matching melodies whose pitch name and sequence size

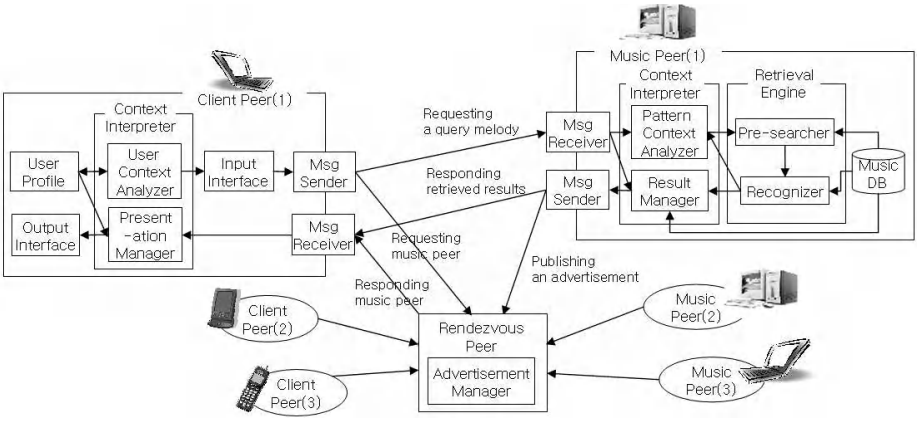


Fig. 2. The architecture of the proposed system

are equal to the query melody. The result analyzer selects the information of the retrieved music according to content items of the query melody and sends it to the client peer. The client peer receives the retrieved result and shows it after it is customized by the presentation manager.

4 Conclusions

In this paper, we show how contextual information can be utilized for CBMIR in a JXTA network. The context interpreter processes the user context, domain context and pattern context. Peers can be both a client and a server and each peer can run a context interpreter to obtain context information. Users can be provided a tailored input interface and result presentation based on their contexts. We are currently implementing the proposed system and experimenting it using a real music database.

References

1. A. K. Dey and G. D. Abowd, Towards a better understanding of context and context-awareness, Technical report GIT-GVU-99-22, Georgia Institute of Technology, 1999.
2. R. Gold and C. Mascolo, Use of Context-Awareness in Mobile Peer-to-Peer Networks, Proceedings of 8tn IEEE Workshop on Future Trends of Distributed Computing Systems, Bologna, Italy, 2001.
3. P. J. Brown and G. J. F. Jones, Exploiting contextual change in context-aware retrieval, Proceedings of the 2002 ACM symposium on Applied computing, 650-656, 2002.
4. B. Traversat, A. Arora, M. Abdelaziz, M. Duigou, C. Haywood, J. C.Hugly, E. Pouyoul, B. Yeager, Project JXTA 2.0 Super-Peer Virtual Network, Sun Micosystems, Inc., May 2003.
5. G. S. Chung, H. C. Yu, C. S. Hawng, A String Searching Method Based on Vowel for Text Editing, Proceedings of the 20th KISS Spring Conference, 755-758, July 1993.
6. C. Wang, J. Li, S. Shi, A Kind of Content-Based Music Information Retrieval Method in a Peer-to-Peer Environment, 3rd International Conference on Music Information Retrieval, 2002.
7. G. Tzanetakis, J. Gao, P. Steenkiste, A Scalable Peer-to-Peer System for Music Content and Information Retrieval, 4th International Conference on Music Information Retrieval, 2003.
8. S. Baumann, Music Similarity Analysis in a P2P Environment, Proceedings of the 4th Euopean Workshop on Image Analysis for Multimedia Interactive Services, April 2003.

A Model of Pervasive Services for Service Composition

Caroline Funk[1], Christoph Kuhmünch[2], and Christoph Niedermeier[2]

[1] LMU Munich, Mobile and Distributed Systems Group, Oettingentrasse 67, 80538 Munich, Germany
caroline.funk@ifi.lmu.de
[2] Siemens AG, Corporate Technology, Otto-Hahn-Ring 6, 81730 Munich, Germany
{christoph.kuhmuench, christoph.niedermeier}@siemens.com

Abstract. We propose a formal definition of a pervasive service model targeting the very dynamic environments typical of mobile application scenarios. The model is based on a requirement analysis and evolves from existing service definitions. These are extended with pervasive features that allow for the modeling of context awareness, and pervasive functionality needed for the dynamic (re-) composition of services. In order to demonstrate its value, we present an implementation of a pervasive service platform that makes use of the service model.

1 Introduction

Pervasive services and pervasive computing are discussed as the next computing paradigm although a generic and formal definition of pervasive services is missing. In this paper we offer a generic and formal definition based on a detailed analysis of the requirements of service (re-) composition.

Recent developments of wireless and sensor network technologies allow for the ad-hoc access to many different computing devices and sensors. Pervasive services make use of these technologies in order to allow for the seamless access to infrastructure and services anywhere, anytime, and in any format. The functionality of a pervasive service can be wide ranging. Typically, it provides access to content or structured information. Examples of content are text, images, or media streaming content, i.e. audio and video. Structured information may be represented by XML documents created from database queries. In addition to content and information provisioning functions, a service may also provide access to hardware devices like printers, displays, speakers, or microphones using many different access protocols. Furthermore, context-aware services like navigation systems require context information.

Often, a single pervasive service is not very useful, e.g. a video decoder is not very useful without a display capable to show the decoded media stream. Thus, to be useful a pervasive service has to support service composition, i.e. it must be able to cooperate with other pervasive services. To accomplish this in a user-friendly way, context information has to be considered while (re-) composing.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 215–224, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Our service model takes these aspects into account and aggregates them in a compact representation. In this way it allows for a formal representation of services and indicates whether two service instances can interact.

Scenario: An important aspect of pervasive services is the composition with other services. Consider two service types S_1 and S_2 that should cooperate in order to build a composite service C together. An example is the consumer-producer concept where S_1 is the producer (e.g. a video server) and S_2 the consumer (e.g. a video display device). During runtime usually multiple instances of S_1 and S_2 can be found. For example, one instance of S_1 is a video server for private videos and a second instance a server for corporate videos. Similarly, the display of a mobile device, the display in the corporate conference room, and the display in the user's office are three instances of S_2.

According to the definition of C – its contextual requirements and rules – we need to select one instance for each service type, i.e. one video server and one display: If only colleagues are present the user wants to show a private holiday video. The user starts the composite service and the private video server and mobile device display are used. Then the user enters the office and he comes into access range of the office display. Since it is larger than the one currently used, a context-aware recomposition is carried out and the private video server is now connected to the office display. In a pervasive world this recomposition is done automatically. A change in context, e.g. the nearing presence of the user's supervisor triggers another recomposition which results in the substitution of the private video server with the corporate one. All the time, the conference room display has been available but wasn't used because the user was not in the conference room.

Structure: In the scenario, different requirements for a pervasive service model can be deduced, and will be explained in Section 2. In Section 3 we give an overview of related research activities. In Section 4 we derive a static service model from the requirements discussed in the previous sections while Section 5 discusses the dynamic model. Section 6 presents the implementation of the model. Finally Section 7 concludes the paper with an outlook to future research.

2 Requirements of a Pervasive Service Model

A user perceives a pervasive service as a homogeneous unit that provides him with all the functionality he needs to accomplish a certain task, whereas the user might not be aware of the dynamics a pervasive system has to deal with. A pervasive service may interact with other services in order to access a maximum of information and functionality and therefore build a composite service with a homogeneous interface to the user. At the same time the context information and the availability of service instances change continuously. For such a dynamic environment that might afford recomposition and reconfiguration, the following four central requirements of a pervasive service can be identified: (1) Selectability, (2) Parameterisability, (3) Inter–Service Dependencies, (4) Context Awareness.

(1) Selectability: The essential characteristic of a service is the *functionality* it provides and thus is the key element for service selection.

Services offer their functionality via *access protocols*. Often competing protocols for the same functionality exist. For example media sessions can be initialised by either using the ITU-T standards H.32x or the IETF Session Initiation Protocol, SIP. Both protocols can be used to access similar functionality but in different ways. Other prominent examples for competing access protocols can be found in the middleware area. Here, CORBA, SOAP, or RMI provide similar functionality. Obviously services need to use the same protocol for communication and thus the protocol is the second selection element.

In order to differentiate between services with the same functionality, *attributes* are required. Attributes may describe a service's quality, its costs, or the identity of its operator. For example, the usage of a service may be bound to certain costs, such as the streaming of music. If one can choose between two music services offering the same functionality, but one service offers songs at a reduced price, the less expensive service should probably be used. While composing the music service with a playback service, this information has to be taken into account.

(2) Parameterisability: A service needs to be adaptable to specific user or situation–specific requirements, i.e. the service model needs to represent service parameters. An example parameter for a display service is the background color.

(3) Inter-Service Dependencies: A service may depend on other services, i.e. it needs input information to be provided by other services and offers output information to them. For example, a service that needs an authentication service in order to verify the users data can only be used for composition if an instance of the authentication service is available.

A service often needs to exchange information with other services. Therefore, it is important for pervasive services to allow for two goals: First, pervasive services need to be *composable*, i.e. two or more single pervasive services can be connected to each other. Second, a composite service needs to be *recomposable*, i.e. the set of currently interacting services needs to be changeable. Service instances need to be replaceable by other service instances during run-time without serious disturbance of the interaction.

In order to be able to carry out recomposition, it is necessary to know the *state* of a service. During recomposition it might be necessary to map or transfer the state of one service to another substitute service. Example: if the connection to a video service degrades, at one point the video service might be replaced by another video service, which provides the same video. Then the new video service needs to be configured in such a way that it continues the video at the position where the old video service quit operation, instead of starting from the beginning.

(4) Context Awareness: Context information needs to be described in two ways: First, a service itself may be context-aware, i.e. it processes context information like location, temperature, etc. Second, a set of composed services may depend

on context information. One may want to use a certain service instance only if certain contextual requirements are fulfilled. Consider an example scenario where a display service is available at a certain location. Of course this service should only be used if the user is in the same room as the display.

3 Related Work

There are different efforts to formalize mobile or pervasive computing and to formalize the service composition process and different aspects thereof.

RM-ODP specifies concepts and a framework for the standardized description of open distributed systems. According to RM-ODP [1] a service is the functionality offered at the interface of an object, where an interface is an abstraction of the behaviour of that object consisting of interactions. The RM-ODP neither models context nor the dynamics of pervasive systems.

[3] extends the RM-ODP with respect to disconnections in nomadic computing and Quality-of-Service requirements. It supports the adaptability of communication and applications with respect to nomadic environments. The dynamics of mobile objects are considered, but (re-) composition aspects are not addressed.

The W3C is working in the field of Web services and defines a service as an abstract resource that represents a capability of performing tasks that form a coherent functionality from the point of view of provider and requester entities [5]. Thus, dependencies of other services or context information are not taken into account. Besides a general service definition the W3C provides a definition of Web services: a software system designed to support interoperable machine-to-machine interaction over a network. The definition of a Web service is bound to different standards and technologies like WSDL, SOAP, or XML [5]. Other standards are not addressed. As well, Web services are defined to be stateless.

The MNM Service Model [4] is a generic model for the management of services. It is derived from a service lifecycle and instantiated for specific services. [4] defines a service as a set of interactions where the identification of the roles of customer, user, and provider is mandatory, but does not take the dynamics of pervasive services into account. Furthermore, [6] have shown that the role model for context-aware services needs to consider the roles of context owner, context provider, and context broker, in order to address context awareness.

4 Formal Definitions for Pervasive Services

In the following, we define a formal model of pervasive services based on the definitions discussed in section 3 and with respect to the requirements in section 2.

We require that a service be selectable and thus needs to have type information. Taking the W3C definition [5] as a starting point, we define a service as an abstract resource providing specific functionalities, made available via specific protocols. Hence the type of a service is defined by the functionalities it provides and the protocols that allow for access to these functionalities. A service S_i depends on a set of functionalities F_i^{in} offered by other services, and it provides

a set of functionalities F_i^{out} to be used by other services. In the video streaming scenario (cf. section 1), the video service requires a display service. Thus a service S_1 can be connected with a service S_2 that provides the functionality required by S_2, i.e. $F_1^{in} \cap F_2^{out} \neq \{\}$ and if they have a common protocol.

Communication protocols typically provide access to the service's functionalities via remote procedure call (e.g. via SOAP, CORBA and RMI) but more application specific protocols like RTSP/RTP, SIP, or H.321 are also valid examples. Obviously a service can only make use of the functionalities of another service if it is provided and used via a communication protocol p_k common to both services. Let P_1^{in} be the set of communication protocols supported by a service S_1 for used functionalities and P_2^{out} be the set of communication protocols supported by a service S_2 for offered functionalities. In that case the two services can only communicate if they have at least one common communication protocol, i.e. $P_1^{in} \cap P_2^{out} \neq \{\}$.

But it is not sufficient to demand that two services share a common communication protocol and functionalities. The desired functionality must be provided and used by this common protocol. Thus we build subsets of functionalities F_{ij}^{in} and F_{ik}^{out} and subsets of protocols P_{ij}^{in} and P_{ik}^{out} where the following must hold for S_i: the functionalities used $F_i^{in} = \bigcup_j F_{ij}^{in}$ and offered $F_i^{out} = \bigcup_k F_{ik}^{out}$ are each the union of all subsets of used and offered functionalities. The sets of communication protocols $P_i^{in} = \bigcup_j P_{ij}^{in}$ and $P_i^{out} = \bigcup_k P_{ik}^{out}$ are each the union of all subsets of protocols.

Based on the previous definitions we are able to build communication paths. A communication path is a pair $(f_{il}^{in}, p_{im}^{in})$ where the functionality $f_{il}^{in} \in F_{ij}^{in}$ and the protocol $p_{im}^{in} \in P_{ij}^{in}$. The same holds for offered functionalities and protocols. Each service S_i has two sets of communication paths: $\{F_{ij}^{in} \times P_{ij}^{in}\}$ and $\{F_{ik}^{out} \times P_{ik}^{out}\}$.

Furthermore, as required in section 2, attributes are needed for selection. An attribute is a read-only element given in a key-value representation, e.g. the display size of a display service. All attributes of a service S_i are combined in the set A_i^r. The attributes, together with the paths built by functionalities and protocols, satisfy requirement (1) for a service to be selectable.

Contrasting with read-only attributes, a parameter that allows a service to be personalised is an attribute that can be modified, e.g. the background color. All parameters of a service S_i are contained in the set A_i^w as key-value pairs. This defines requirement (2), and is a characteristic of pervasive services.

In order to satisfy the criteria of inter-service dependencies (requirement (3)), we make use of the notion of used functionality. Each service that is needed by service S_i can be modeled as a used functionality, and therefore is an element of F_i^{in}.

For the substitution of a service during recomposition it is necessary to have information about the state of a service. During the process of recomposition the state of one service may be transferred to another service in order to be able to fulfill service dependencies. All information about states is held in the set of states Z_i.

As stated in requirement (4), many pervasive services are context aware. Context is defined as information provided by a context information service (CIS). A context aware service S_j requires context information as input, thus we can represent context awareness using input functionality F_j^{in} as defined above.

Summarising the discussion in the previous paragraphs we define a **pervasive service**

$$S_i = (\{F_{ij}^{in} \times P_{ij}^{in}\}, \{F_{ik}^{out} \times P_{ik}^{out}\}, A_i^r, A_i^w, Z_i).$$

According to this definition the **type of a service** is defined by the functionalities it uses and offers via communication protocols, its attributes, parameters, and states. This definition allows us to include a wide field of services including virtual services (e.g. a currency translator) and hardware devices (e.g. a display service of a display or a printing service of a printer). A **service instance** is a concrete implementation of a service type.

A characteristic of a pervasive service is that it's composable. To be able to build a composite service a set $S = \{S_0 \dots S_n\}$ of service types is needed. Furthermore a recipe R describes all criteria and preferences for service instance selection. In particular context information is used, to decribe what service instances will be used at runtime. Moreover the recipe specifies in what way the services have to be parametrised and connected. Toghether, the service types and the recipe form a **composite service** $C = \{S, R\}$. A **(composite service) session** is an instance of a composite service, but often a session is meant when one talks about a composite service.

The **composition** is carried out using the service descriptions, contextual requirements, constraints and preferences from the recipe. First, service instances can be identified and selected. For each required service type, the recipe defines the parameter settings that have to be mapped to the parameters of the service instance. Finally the recipe includes a service digraph that describes the interconnection between the services. Two services S_1 and S_2 can be connected if there exists a path $(f_{1j}^{in}, p_{1r}^{in}) \in S_1$ and a path $(f_{2k}^{out}, p_{2s}^{out}) \in S_2$ where $p_{1r}^{in} = p_{2s}^{out}$ and f_{1j}^{in} and f_{2k}^{out} correspond. As one service may have many paths it may as well be connected to more than one other service. Therefore, one service may have different roles in different connections.

If any parameters or requirements for the session (e.g. a service is not reachable any more) or the composite service (e.g. a new service appears that is better than a used one) change, **recomposition** is necessary. During recomposition all the steps described above need to be carried out again, but with respect to the current state of the session. If services are replaced, the state needs to be transferred in order to continue operation without interruption. Besides service replacement, service reconfiguration may be necessary. This satisfies the requirements of a dynamic environment.

5 The Dynamic Process of Service Composition

Based on the formal model given in the previous section, we want to discuss the process of service composition on the basis of the service lifecycle. In order

to instantiate a composite service, service instances of all relevant service types have to be selected, allocated, parametrized, and connected. In the following, we want a more detailed view on how a service that is a part of a session behaves and which properties it has. In particular we want to discuss the lifecycle and states of a composable service.

5.1 Pervasive Service Lifecycle

Service lifecycle: Fig. 1 shows the lifecycle of a service regarding the composition process in the form of a state diagram. The state diagram of a service viewed without composition may have the states *closed*, *open*, *running*, *interrupted*. While a service in the closed state is only created, in the state open it's already allocated. A *running* service is avtively providing functionality. During this phase it may be interrupted by some event and therefore change its state to *interrupted*.

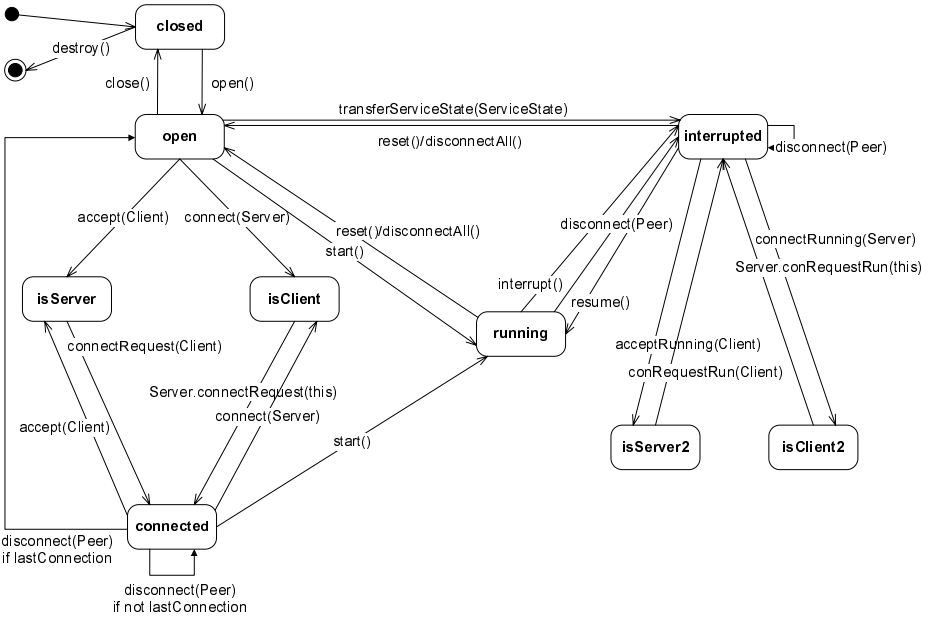


Fig. 1. Lifecycle of a pervasive service

Composition: In order to describe the process of composition we introduce the states *isClient*, *isServer* and *connected*. During the process of connecting two services one composable service passes either the state *isClient*, while waiting for the corresponding service that offers functionality to accept the connection request, or it passes the state *isServer*, while waiting for a service that wants to use offered functionality to connect. The naming *isClient* and *isServer* is chosen to distinguish between used and offered functionality. After being connected to

a peer service (state *connected*), another connection to another peer may be established. This process reflects the formal definition given in section 4 where a path is described between a service that offers functionality and a service that uses functionality. It should be noted that error states and related transitions are not taken into account in fig. 1.

Recomposition: During the process of recomposition, new connections may have to be established. Recomposition may be necessary due to some exception or event, e.g. disconnection of a peer service, a change of context, or a deliberate interrupt. If this happens, a transition from state *running* to state *interrupted* is triggered. In state *interrupted* new connections can be established. To be able to differentiate between connections that are established during (first) composition or during recomposition, the states *isServer2* and *isClient2* are introduced similarly to *isServer* and *isClient*. If a service that is new and hasn't been running yet within the session has to be connected to a formerly running service, the state *interrupted* will be used as well. The new service makes a transition from state *open* to state *interrupted*. This allows for the configuration of a service in order to be able to connect to a formerly running service that has a different internal service state (cf. section 4).

5.2 Interfaces of a Pervasive Service

In order to make a service composable it needs to implement a specific composition interface. This interface allows for the selection, allocation, parametrisation, and connection of the service. The main methods of this interface can be derived from the pervasive service lifecycle shown in the preceding section and won't be explained in more detail here. These interfaces have been used for the prototype implementation shown in the following section.

6 Practical Evaluation

In order to prove the feasibility of our service model we implemented a framework that provides a pervasive service platform (PSP) by making use of the service model described in the previous sections. The platform, partly based on concepts elaborated in the EU project "Daidalos", provides a run-time environment for services that follow the service model. Using this run-time environment, we implemented the scenario described in section 1.

The PSP offers two main elements: (1) It offers an implementation of our service model in terms of interfaces and base classes that provide a skeleton for the development of pervasive services. (2) It provides enabling platform services which are implemented by making use of the skeletons.

Fig. 2 shows the elements of our architecture. There are six main enabling platform services: The context management service incorporates context information services (CIS). The personalisation service takes care of user preferences, that are guarded by security and privacy services. The service discovery service performs all tasks to provide the service manager with services. The service

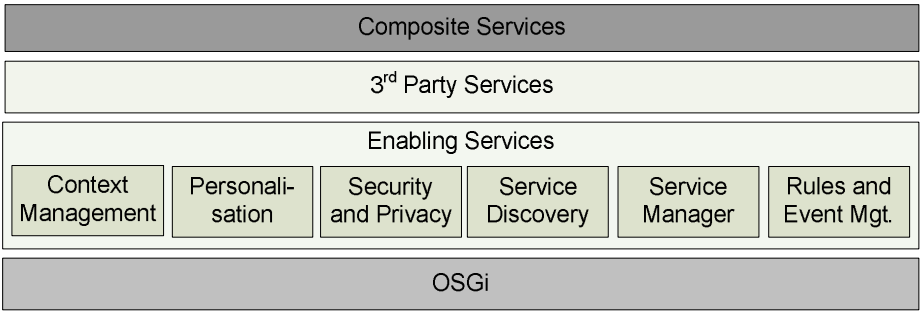


Fig. 2. Structure of the example implementation

manager is the main service that is responsible for the composite service lifecycle management, while using the rules and event management service.

On top of the enabling services, pervasive services are implemented using the service model, e.g. a video service and a display service. These so-called 3rd party services are combined by the service manager into composite services that are used by the user.

We realised the platform implementation using the programming language Java and the Open Service Gateway Initiative's specification [2] in its current version (3.0). OSGi defines a standardised, component-oriented computing environment for networked services. The basis of OSGi is an arbitrary combination of a hardware platform and operating system that runs the OSGi execution environment, i.e. a Java Virtual machine. It allows for the management of services, which includes the hot installation, update, and removal of services. Furthermore, the framework allows for the dynamic discovery of services and provides a number of other useful features that simplify the development of a pervasive service platform. More information on OSGi can be found in [2].

7 Conclusions

In this paper, we elaborated the concept of pervasive service composition. In our analysis of the prerequisites for a pervasive service model, we identified and described four key aspects: (1) Selectability of services via characteristic attributes, (2) Parameterisability allowing for dynamic adaptivity of services, (3) Inter-Service Dependencies facilitating composition and recomposition of services, and (4) Context Awareness as a major attribute of pervasive services as well as a tool for dynamically optimised service composition. Two important steps on the way to realization of the proposed concept were taken: First, a formalism for characterisation of pervasive services in terms of input and output functionalities as well as associated communication protocols has been proposed. Second, a model for the life cycle of pervasive composable services has been elaborated. A first implementation of the proposed service model based on the OSGi framework has also been presented.

In order to further establish the proposed concept we plan the following steps: (1) Design and evaluation of a description language for the recipe describing the structure of composed services; (2) Further elaboration of the service lifecycle, in particular regarding the transfer of a service's state between two equivalent services as well as concepts for handling of faults occurring during service composition or recomposition. By realizing non-trivial composite pervasive services we will be able to evaluate the validity of our concept for practical purposes.

Acknowledgements

The authors thank the members of the Mobile and Distributed Systems Group, a team of researchers directed by Prof. Dr. Linnhoff-Popien at the University of Munich (www.mobile.ifi.lmu.de) for valuable comments.

The work presented in this paper has partially been funded by the EU project IST-2002-506997 "Daidalos" (www.ist-daidalos.org).

References

1. Open distributed processing – reference model – part 2: Foundations international standard 10746-2 itu-t recommendation x.902, 1995.
2. O. Alliance. Osgi service platform release 3. Technical report, 2003.
3. A. B. et.al. An information model for nomadic environments. In *DEXA '98: Proceedings of the 9th International Workshop on Database and Expert Systems Applications*, Washington, DC, USA, 1998. IEEE Computer Society.
4. M. e. Garschhammer. Towards generic service management concepts – a service model based approach. In *7th International IFIP/IEEE Symposium on Integrated Management (IM 2001), Seattle, Washington, USA*, May 2001.
5. W. W. Group. Web services glossary, 2004.
6. H. Hegering, A. Kpper, C. LinnhoffPopien, and H. Reiser. Management challenges of contextaware services in ubiquitous environments. In *SelfManaging Distributed Systems; 14th IFIP/IEEE Int. Workshop on Distributed Systems: Operations and Management, DSOM 2003, Heidelberg, Germany*, 2003.

Selection Using Non-symmetric Context Areas

Diane Lingrand, Stéphane Lavirotte, and Jean-Yves Tigli

I3S, UNSA/CNRS UMR 6070, B.P. 121, F06903 Sophia Antipolis, France
lingrand@i3s.unice.fr
http://www.i3s.unice.fr/~ lingrand/

Abstract. This paper targets with applications running on mobile devices and using context informations. Following previous studies from other authors, we extend the notion of context area replacing distance function by cost function. Using this extension, we exhibit three different modes of selection and demonstrate their differences on a mobile applications: the museum visit.

1 Motivations

With the expansion of mobile devices in our life (PDA, cellular phones, ...), we have observed for some years the development of applications taking context into account. For example, it is natural to take into account the localization of a user in an application which aims at giving a list of nearby restaurants. Moreover, the application should also consider the opening days and hours in order to select relevant items in this list using the current date and time. Context-aware applications are intended to simplify the interface between the user and the machine.

The concept of context and its evaluation are often redefined depending on application needs [3,4,5,6]. However, some authors propose a general context definition such as Dey [2]: *"Context is any information that can be used to characterize the situation of an entity"*. In order to use context informations, Pauty, Couderc and Banâtre [1] propose a definition and an evaluation of context using distance functions.

In this paper, we come back on previous definitions of context, context area and selection modes using distances. We exhibit an example where these definitions of context and context area are not satisfying. We then propose a new formalism to define context and context area. Using this formalism, we introduce again the selection modes definitions that are, in this case, not symmetric. We illustrate this on a mobile applications: the museum visit.

2 Context Model

2.1 Context and Context Area Definitions

The context space $\mathcal{E}$ is defined as a state space composed by several contextual components e_i. Each component is bound to a distance function d_i:

$$\mathcal{E} = \{\{e_1, ..., e_n\}, \{d_1, ..., d_n\}\}$$

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 225–228, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Several authors [9,8] have made a classification of the different types of components in several families: environmental context, user context, computer context and time context.

In the context space are **context instances** for which the components e_i are taking values in a determined set.Assuming that the distance function in the context space is well defined, we need to define the notion of proximity that is the context area or neighborhood of a context instance E:

$$Z(E) = \{F \mid d(F, E) \leq D\} \text{ , where D is a constant.} \tag{1}$$

2.2 Discussion on Distances

Do we really need to extend the constant D to a function? In [1], D is replaced by a function of E and F. Replacing the constant D by a function has the advantage of permitting the expression of a large variety of constraints. However, it is a too broad definition that needs to be restricted to make sense.

Is it necessary to be a distance? Let us take the example of a hiker in the mountain: going from A (in the valley) to B (the summit) may be more costly (in term of effort or gasoline) than going from B to A. The cost, in this example, is not symmetric: this is not a distance!

However, we need to be able to compare costs between several contexts and to determine neighborhoods. We now show how to replace a cost function instead of distance function in our formalism.

2.3 Introducing a Cost Function

We define the cost function by the following properties:

$$c(x, y) \geq 0 \text{ , } c(x, y) \neq 0 \Rightarrow x \neq y \text{ and } c(x, y) \leq c(x, z) + c(z, y) \tag{2}$$

These properties are similar to the properties of a distance function except the symmetry. The context space is redefined as follow: $\mathcal{E} = \{\{e_1, ..., e_n\}, \{c_1, ..., c_n\}\}$ where the c_i are cost functions, and the context area by:

$$Z(E) = \{F \mid c(F, E) \leq C\} \tag{3}$$

where c is a cost function defined by equation 2 and C is a constant cost.

With this definition, loosing the symmetry of the distance function, we also loose the symmetry of the context area as we will study in the next paragraph.

2.4 Selection Mode

Knowing a context instance, one may want to select all context instances near E. There are two ways of doing that: (i) select the context instances in the context area of E (**endo selection**) or (ii) select the context instances which context area contains E (**exo selection**). If we are interested in both types of selection, we can use the **bilateral selection**:

$$S_{\text{bilateral}}(E) = S_{\text{endo}}(E) \cap S_{\text{exo}}(E)$$

Considering definition (1), there is no difference between endo and exo selection, as said in [1]. But considering definition (3), the endo and exo selection modes are different because of the lack of symmetry. This leads to interesting properties in the selection. We will now illustrate this.

3 Experimentation: The Museum Visit

In this well-known application, each visitor has a PDA for commenting the pictures displayed. When the visitors PDA detects a picture nearby, it displays informations on this picture. However, since the hall may be large and the pictures high, we also consider the fact that the visitor may have a picture in his back nearer than the picture he is looking at. Of course, in this case, he would like to have the information on the picture he is looking at rather than the one behind him. This is an example showing that we need to select pictures regarding the distance between picture and visitor and the orientation of the visitor.

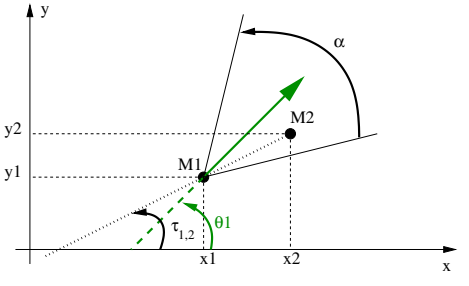


Fig. 1. Notations for the context space in the case of the museum visit. The point M_1 represents the visitor and M_2 the picture. M_1 is defined by its 2D position (coordinates x_1 and x_2) and its orientation given by the angle θ_1. The direction of the user with respect to the picture $\tau_{1,2}$ is centered on θ_1 with amplitude α.

The context space is composed by a 2D position $(x; y)$ and an orientation θ. We define the context area limiting the Euclidean distance to D and the angle variation to $\frac{\alpha}{2}$ (see figure above). The cost between points M_1 (the visitor) and M_2 (the picture) is then given by:

$$c([x_1, y_1, \theta_1]^T, [x_2, y_2, \theta_2]^T) = \max\left(\frac{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}{D}, \frac{|\tau_{1,2} - \theta_1|}{\alpha/2}\right)$$

where $\tau_{1,2} = (\widehat{Ox, \overrightarrow{M_1 M_2}})$ measures the orientation of the vector $\overrightarrow{M_1 M_2}$.

The context area is defined by: $c([x_1, y_1, \theta_1]^T, [x_2, y_2, \theta_2]^T) \leq 1$.

As seen in figure 2, the visitor must see the picture: this is the endo selection. But if the visitor is behind the picture, he cannot see this picture. Using the exo selection, we select users in front of the picture but not necessary looking at the picture. The correct selection is the bilateral selection: the visitor must see the picture and the picture must have the visitor in his field of view.

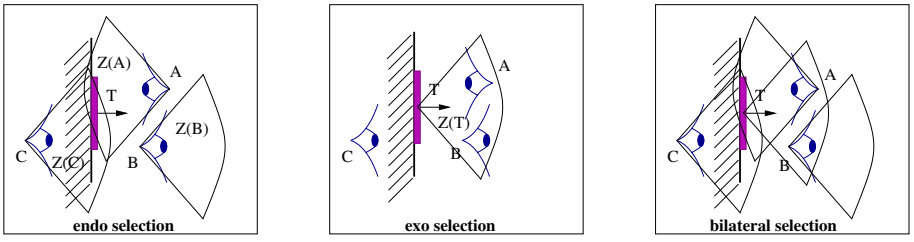


Fig. 2. The three different modes of selection. The selection we need in this application is the bilateral selection.

4 Conclusion

In this paper, we have extended the context area definition using cost functions instead of distance functions. This allows to express a larger family of applications and leads to different selection modes: endo, exo and bilateral.

We have applied this context formalism to the well-known example of the museum visit considering both location and sight view of the visitor.

Future work will focus on dynamic cost composition in order to adapt our system to the availability of different context components.

References

1. Pauty, J., Couderc, P., Banâtre, M.: Synthèse des méthodes de programmation en informatique contextuelle. Technical Report 1595, IRISA (2004)
2. Dey, A.K.: Understanding and using context. Personal and Ubiquitous Computing **5** (2001) 4–7
3. Want, R., Hopper, A., Falcão, V., Gibbons, J.: The active badge location system. ACM Transactions on Information Systems (1992)
4. Schilit, B.N., Hilbert, D.M., Trevor, J.: Context-aware communication. IEEE Wireless Communications **9** (2002) 46–54
5. Long, S., Kooper, R., Abowd, G.D., Atkeson, C.G.: Rapid prototyping of mobile context-aware applications: the cyberguide case study. Int. Conf. on Mobile Computing and Networking, White Plains, NY, ACM Press. (1996) 97–107
6. Abowd, G.D., Atkeson, C.G., Hong, J., Long, S., Kooper, R., Pinkerton, M.: Cyberguide: A mobile context-aware tour guide. Wireless Networks **3** (1997) 421–433
7. Strang, T., Linnhoff-Popien, C.: A context modeling survey. In Workshop on Advanced Context Modeling, Reasoning and Management associated with the 6th UbiComp, Nottingham, England (2004)
8. Coutaz, J., Rey, G.: Foundations for a theory of contextors. 4th Int. Conf. on Computer-Aided Design of User Interfaces, Valenciennes, France (2002) 283–302
9. Chen, G., Kotz, D.: A survey of context-aware mobile computing research. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College (2000)

Sharing Context Information in Semantic Spaces

Reto Krummenacher, Jacek Kopecký, and Thomas Strang

Digital Enterprise Research Institute, University of Innsbruck, Austria

Abstract. In a highly mobile and active society, ubiquitous access to information and services is a general desire. The context of users is of high importance for an application to adapt automatically to changing situations and to provide the relevant data. We present a combination of space-based computing and the Semantic Web to provide a communication infrastructure that simplifies sharing data in dynamic and heterogenous systems. The infrastructure is called *Semantic Spaces*.

1 Introduction

In general, context-aware applications have to deal with big volumes of heterogenous data from various heterogenous data sources. Moreover, most of the available IT applications depend on synchronous communication between information sources and sinks.

Tuple Spaces [3] allow distributed sharing of information by various devices without requiring synchronous connections. The Semantic Web [1] extends the Web with machine-processable semantic data, allowing data exchange in heterogeneous application fields. Combining the two introduces a new communication platform that provides persistent and asynchronous dissemination of machine-understandable information [2], especially suitable for distributed services. We call this combined communication platform *Semantic Spaces*.

In Section 2 we first present a context-aware application and show how it would be implemented without Semantic Spaces. Section 3 depicts the technical background for our approach, while Section 4 talks about how the application from section 2 benefits from using Semantic Spaces. Finally in Section 5 we conclude the paper.

2 Scenario: Enriching the Conference Experience

To illustrate our interaction scheme and how it would improve a ubiquitous computing system, we present a hypothetical use case: a mobile application supporting conference participants. The application enriches the conference experience and improves communication between attendees.

At the Conference: As Alice arrives at the conference venue, the application completes the registration process. The application has to query the necessary information about Alice from her *user profile service* (UPS) and forward it to the *registration service*. After the first presentation, Alice has to change buildings. There is a problem with the central heating in the second building indicated by

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 229–232, 2005.
© Springer-Verlag Berlin Heidelberg 2005

the *building management service* (BMS). In consequence the application informs Alice that it will be unusually cold. The provision of the relative temperature is a non-trivial task. The application and the BMS have to agree ex ante on the "value" for cold.

Coffee breaks and lunch time (time and location are obtainable from the *conference program service* (CPS)) are very important for successful social networking, and the device guides Alice to the people whom she wants to meet. To be able to provide this simple service the application first retrieves the set of people she would like to meet from her UPS. In order to discover all the desired fellow researchers the application exchanges various messages with the CPS and the *location service* (LoS).

Our application integrates many different services in order to automatically adapt to changing situations. Many messages are sent around and the information contained is often encoded in different ways by different sources. Hence, the application has to deal with heterogenous data communicated over heterogenous channels.

3 Technical Background

This section gives a short introduction to the two fundamental technologies underlying the Semantic Spaces idea: Tuple Spaces and the Semantic Web.

Tuple Spaces were first introduced in the mid 80's in form of the Linda programming language [3]. Linda programs exchange data by using Tuple Spaces as shared memory for data tuples. A tuple is an ordered set of typed parameters. Linda defines a simple interface to access the space: out(tuple) to write and in(template) to read from the space, where templates match tuples that have the specified internal structure. The template matching procedure supports neither inference nor advanced querying — these features are added by the integration of Semantic Web technology.

In the **Semantic Web**, data is represented in the Resource Description Framework (RDF) or in languages based on it. Semantic data representation together with reasoning engines permits information from various sources (possibly using differing vocabularies) to be easily combined, and further facts inferred; going beyond subsumption reasoning of common domain classifications.

In contrast to the current synchronous communication approaches, Semantic Spaces decouple the information sources and clients in the following ways:

- **Time autonomy:** every agent can access the space at its own discretion.
- **Reference autonomy:** agents do not need to know each other, they only need to know the shared space.
- **Vocabulary autonomy:** by using semantic mappings between vocabularies, at least partial understanding of heterogenous data is achieved.

4 Applying Semantic Spaces to the Scenario

It is the central functionality of our context-aware application to collect and process vast amounts of information about Alice and the conference. In partic-

ular the application makes use of time and location information, temperature sensors and user profiles. The key processes here are the discovery of information sources on the one hand and the collecting and dissemination of information on the other. Most current mechanisms rely heavily upon a priori identification of all the things one would want to communicate with [4]. Our goal is to automate these key processes and to run them without human intervention. Sharing basic vocabularies and allowing machines to understand and reason about the semantics of context information enables them to interoperate automatically, i.e., to use, interpret and combine context information, as well as to infer further facts [5]. Below we apply Semantic Spaces to the conference scenario to demonstrate how the coordination of information providers and consumers can be improved.

At the Conference: As Alice arrives at the conference and her device stores her profile in the space, the registration service has all the necessary information to automatically complete Alice's registration. The presentation venue, attendee locations and profiles are described by semantic data in the space and hence every application interested in any of that information can simply read it from the space and does not need to discover the appropriate services and interact with them. When Alice needs to change the building, the application informs her about the unusually low temperature — revealed by sensors — in the second building. To do so, the application combines and reasons about the sensor data, location information and Alice's temperature preferences.

During the breaks the application assists Alice in finding the fellow researchers she desires to meet. All the necessary information (her location, the information about the people to meet and their locations) can of course be found in the space. Every application involved in the organization of the conference can read this information from the space and many-to-many message exchanges are not required.

The following is a sample of triple data present in the space that is used to bring Bob and Alice together during the coffee break:

```
<Alice wants-to-meet Bob>       - from Alice
<Alice is-at room-xyz>          - from location service
<Bob is-at room-zyx>            - from location service
<coffeebreak at-time 10.30am>   - from conference organizer
<coffeebreak is-at room-xyz>    - from conference organizer
```

Context-aware applications demand an exhaustive volume of communication to ensure the coordination and data exchange between information providers and consumers. Semantic Spaces replace the manifold message exchanges by read and write operations on a commonly accessible space. Adding semantics to the space reduces data heterogeneity and ambiguity, as well as incomplete and sometimes low quality data. Reasoning, knowledge inference and data mediation are effective tools of the Semantic Web and can be performed directly in the space.

5 Conclusions and Future Work

The Semantic Web aims at making information understandable to computers by adding machine-processable semantics to data. Tuple Spaces additionally allow complex networks of message exchanges to be replaced by simple read and write operations in a commonly accessible space. Overlaying the combination of Semantic Web and Tuple Spaces on a ubiquitous computing environment allows for interlinking services and sensors with clients without an explosion of communication. Semantic Spaces provide not only a promising basis for an infrastructure that addresses data and communication heterogeneity, but certainly an infrastructure that largely decreases the communication overhead commonly necessary to gather context information.

The proposed Semantic Spaces are not intended to replace all communication and coordination methods in distributed applications. Synchronous point-to-point links are still going to be used for time-critical tasks where one node needs to interrupt another. In spaces, a node will not notice an urgent request if it does not check the content often enough. Notification mechanisms are proposed for such situations, however bypassing the space is most likely the better solution.

When concretizing the architecture for the space infrastructure, we need to take into consideration distribution and scalability issues, as we are mainly dealing with mobile networks that are at once highly dynamic and weakly connected. Other immediate concerns are security and trust. Sharing information, particularly about humans, is a delicate task.

Acknowledgements. The work is funded by the European Commission under the projects DIP, KnowledgeWeb, InfraWebs, SEKT, and ASG; by Science Foundation Ireland under the DERI-Lion project; by the Austrian Federal Ministry for Transport, Innovation, and Technology under the FIT-IT projects RW^2 and TSC; and by NCR Korea.

References

1. Berners-Lee T., Hendler J. and Lassila 0., 2001: The Semantic Web, The Scientific American, May 2001.
2. Fensel D., 2004: Triple-space computing: Semantic Web Services based on persistent publication of information, Proc. of the IFIP Int'l Conf. on Intelligence in Communication Systems, Bangkok, Thailand: 43-53.
3. Gelernter D., 1985: Generative Communication in Linda. ACM Transactions on Prog. Languages and Systems, 7(1): 80-112.
4. Lassila O., and Adler M., 2003: Semantic Gadgets: Ubiquitous Computing Meets the Semantic Web. In D. Fensel et al.: Spinning the Semantic Web: 363-376.
5. Tan J.G., Zhang D., Wang X. and Cheng H.S., 2005: Enhancing Semantic Spaces with Event-Driven Context Interpretation. Proc. 3rd Int'l Conf. on Pervasive Computing, Munich, Germany: 80-97.

GADA 2005 PC Co-chairs' Message

We wish to extend a warm welcome to GADA'05, The Second International Workshop on Grid Computing and its Application to Data Analysis, held in Ayia Napa (Cyprus), in conjunction with the On The Move Federated Conferences and Workshops 2005 (OTM'05).

This time around we have been fortunate enough to receive an even larger number of highly informative and engaging papers from all corners of the globe covering a wide range of scientific subjects and computational tool designs. It has been very inspiring to observe the extensive and well-formulated work being done in the development of processing and resource management tools to facilitate the ever increasing computational requirements of today's and future projects.

Empowering scientists and systems end-users with intuitive technical tools is an important part of the bridge connecting our high-tech innovations with the classical sciences. This year, the move to a web service oriented Grid approach seems the more apparent. An important step in providing heterogeneous and compatible resources across this research community. Nonetheless, we believe there is still much work to be done in easing the task of data analysis. This realm might include more intuitive interfaces, management of distributed storage resources, subsequent data mining, and so on. We seem to be able to tackle any analytical, numerical or storage problem efficiently, but there remains a gap in the access to these solutions, perhaps one of human-computer interaction.

Our foremost goal is to continue to create and improve upon a forum for the interchange of ideas and experiences in relevant areas within the scientific and commercial GRID community.

The set standard was higher than our expectations and, although we are constrained to include a relatively small subset of paper submissions within this vast field, this year the workshop received 59 submissions from which the 18 papers making up the technical programme were selected. We hope that our selection represents a good overview of high-quality material and well conceived ongoing research.

This workshop could not have taken place without considerable enthusiasm, support and encouragement as well as sheer hard work. We would like to thank to the GADA'05 Program Committee who gave their time and energy to ensure that the conference maintains high technical quality and runs smoothly. In the same way, we are also especially grateful to the organizers of the OTM'05 conferences for the support and encouragement they extend to this workshop. The close cooperation between GADA'05 and the OTM'05 organization allows us to contribute to the growth of this research community.

August 2005

Pilar Herrero, Universidad Politécnica de Madrid
María S. Pérez, Universidad Politécnica de Madrid
Victor Robles, Universidad Politécnica de Madrid
Jan Humble, University of Nottingham
(GADA'05 Program Committee Co-Chairs)

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, p. 233, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Coordinated Use of Globus Pre-WS and WS Resource Management Services with Grid *W*ay*

Eduardo Huedo[1], Rubén S. Montero[2], and Ignacio M. Llorente[2,1]

[1] Laboratorio de Computación Avanzada, Simulación y Aplicaciones Telemáticas, Centro de Astrobiología (CSIC-INTA), 28850 Torrejón de Ardoz, Spain
[2] Departamento de Arquitectura de Computadores y Automática, Universidad Complutense, 28040 Madrid, Spain

Abstract. The coexistence of different Grid infrastructures and the advent of Grid services based on Web Services opens an interesting debate about the coordinated harnessing of resources based on different middleware implementations and even different Grid service technologies. In this paper, we present the loosely-coupled architecture of Grid *W*ay, which allows the coordinated use of different Grid infrastructures, although based on different Grid middlewares and services, as well as a straightforward resource sharing. This architecture eases the gradual migration from pre-WS Grid services to WS ones, and even, the long-term coexistence of both. We demonstrate its suitability with the evaluation of the coordinated use of two Grid infrastructures: a research testbed based on Globus WS Grid services, and a production testbed based on Globus pre-WS Grid services, as part of the LCG middleware.

1 Introduction

Since the late 1990s, we have witnessed an extraordinary development of Grid technologies. Nowadays, different Grid infrastructures are being deployed within the context of growing national and transnational research projects. The majority of the Grid infrastructures are being built on protocols and services provided by the Globus Toolkit (GT) [1], becoming a *de facto* standard in Grid computing.

The coexistence of several projects, each with its own middleware developments, adaptations, extensions and service technologies, give rise to the idea of coordinated harnessing of resources, or contributing the same resource to more than one project. Moreover, the advent of GT4 and the implementations of Grid services as Web Services by following the WSRF (WS-Resource Framework) [2], arises the idea of a gradual migration from pre-WS Grid services to WS ones, and even, the long-term coexistence of both types of services.

Instead of tailoring the core Grid middleware to our needs (since in such case the resulting infrastructure would be application specific), or homogenizing

* This research was supported by Ministerio de Educación y Ciencia, through the research grant TIC 2003-01321, and by Instituto Nacional de Técnica Aeroespacial "Esteban Terradas" – Centro de Astrobiología. The authors participate in the EGEE project, funded by the European Union under contract IST-2003-508833.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 234–243, 2005.
© Springer-Verlag Berlin Heidelberg 2005

the underlying resources (since in such case the resulting infrastructure would be a highly distributed cluster), we propose to strictly follow an "end-to-end" principle. In fact, Globus architecture follows an hourglass approach, which is indeed an "end-to-end" principle. In an "end-to-end" architecture, clients have access to a wide range of resources provided through a limited and standardized set of protocols and interfaces. And resources provide their capabilities through the same set of protocols and interfaces. In the Grid these are provided by the core Grid middleware: Globus in this case. Just as, in the Internet, they are provided through the TCP/IP set of protocols.

One approach is the development of gateways between different middleware implementations [3,4]. Another approach, more in line with the Grid philosophy, is the development of client tools that can adapt to different middleware implementations. If we consider that nearly all current projects use Globus as basic Grid middleware, it could be possible a shift of functionality from resources to brokers or clients. This would allow to access resources in a standard way, making the task of sharing resources between organizations and projects easier.

The aim of this paper is to present and evaluate a loosely-coupled architecture that allows the simultaneous and coordinated use of both pre-WS and WS GRAM services, as well as other Grid services. The rest of the paper is as follows. Section 2 compares pre-WS Grid services with WS ones. Section 3 introduces the Globus approach for resource management. Section 4 introduces the Grid*W*ay approach for job management. Section 5 shows some experiences and results. Finally, Section 6 ends up with some conclusions.

2 From Pre-WS to WS Grid Services

The main reason behind the moving from pre-WS to WS Grid services is that, according to the Grid's second requirement proposed by Foster [5], a grid must be built using standard, open, general-purpose protocols and interfaces. However, many people is still reluctant to this change because it could bring an important performance loss. In fact, the Grid's third requirement is that a grid must deliver nontrivial qualities of service, in terms of response time, throughput, security, reliability or the coordinated use of multiple resource types.

On one hand, pre-WS Grid services are based on proprietary interfaces (although usually implemented over standard protocols, like HTTP). On the other hand, WS Grid services are based on the WS-Resource Framework (WSRF) [2], a standard specification fully compatible with other Web Services specifications. In fact, WSRF can be viewed as a set of conventions and usage patterns within the context of established Web Services standards, like WS-Addressing.

WSRF defines the WS-Resource construct as a composition of a Web Service and a stateful resource. The Open Grid Services Infrastructure (OGSI) was previously conceived as an extension of Web Services to have stateful WS-Resources [6]. However, the implementation of OGSI resulted in non standard, complex and heavy-weight Grid services. Moreover, it jeopardized the convergence of Grid and Web Services. On the contrary, Grid services implemented

as Web Services are easier to specify and, therefore, to standardize. Thus, WS Grid services provide a way to construct an Open Grid Services Architecture (OGSA) [7] where tools from multiple vendors interoperate through the same set of protocols and interfaces, implemented in different manners.

3 The Globus Approach for Resource Management

The Globus Toolkit [1] has become a *de facto* standard in Grid computing. Globus services allow secure and transparent access to resources across multiple administrative domains, and serve as building blocks to implement the stages of Grid scheduling [8]. Resource management is maybe the most important component for computational grids, although it could be also extended to other non-computational resources. The Globus Resource Allocation Manager (GRAM) [9] is the core of the resource management pillar of the Globus Toolkit.

In pre-WS GRAM (see Figure 1), when a job is submitted, the request is sent to the Gatekeeper service of the remote computer. The Gatekeeper is a service running on every node of a Globus grid. The Gatekeeper handles each request, mutually authenticating with the client and mapping the request to a local user, and creates a Job Manager for each job. The Job Manager starts, controls and monitors the job according to its RSL (Resource Specification Language) specification, communicating state changes back to the GRAM client via callbacks. When the job terminates, either normally or by failing, the Job Manager terminates as well, ending the life cycle of the Grid job.

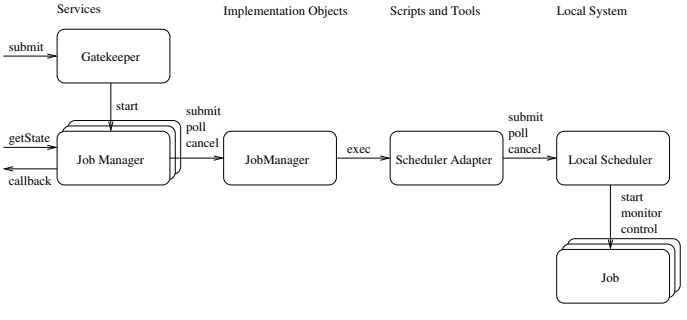


Fig. 1. Architecture of the pre-WS Globus Resource Allocation Manager (GRAM)

In WS GRAM (see Figure 2), when a job is submitted, the request is sent to the Managed Job Factory service of the remote computer. The Managed Job Factory and Managed Job are two services running on every node of a Globus grid. The Managed Job Factory handles each request and creates a Managed Job resource for each job. Authentication is performed via Web Services mechanisms and some operations are mapped to a local user via `sudo`. The Managed Job service uses a Job Manager to start and control the job according to its

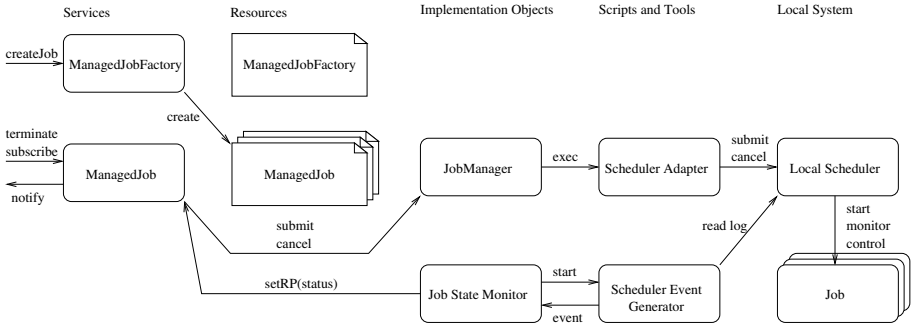


Fig. 2. Architecture of the WS Globus Resource Allocation Manager (GRAM)

RSL specification, mapping the request to a local user and communicating state changes back to the GRAM client via WS-Notifications [10]. When the job terminates, either normally or by failing, the Managed Job resource is destroyed, ending the life cycle of the Grid job.

Although the use of Web Services entails some overhead, the implementation of WS GRAM has been optimized in several ways. For example, it provides better job status monitoring mechanism through the use of a Job State Monitor (JSM), which in turns uses a Scheduler Event Generator (SEG), instead of implementing a polling mechanism in the Job Manager, as in pre-WS GRAM. It also provides a more scalable/reliable file handling through the use of a Reliable File Transfer (RFT) service instead of the `globus-url-copy` command used directly by the Job Manager in pre-WS GRAM. Moreover, WS GRAM only supports GridFTP for file transfer and the use of GASS (Global Access to Secondary Storage) caching has been removed. In any case, WSRF-based Grid services in GT4 clearly outperforms heavy-weight OGSI-based Grid services in GT3 [11].

As can be seen in Figure 2, WSRF separates services, resources and implementation objects. This way, it is easier to standardize a service architecture, like OGSA, since only services and resource properties representing resource state have to be specified in the standardization documents.

GRAM operates in conjunction with a number of schedulers including Condor, PBS and a simple "fork" scheduler. The Job Manager provides a plugin architecture for extensibility. When the Job Manager is respectively invoked by the Gatekeeper or Managed Job to process a job request, it maps the request to a local scheduler. These plugins provide a set of programs and scripts that map job requests to scheduler commands such as submit, poll or cancel.

4 The Grid *W*ay Approach for Job Management

Grid *W*ay is a job management system, whose main objective is to provide a decentralized, modular and loosely-coupled architecture for scheduling and executing jobs in dynamic Grid environments. The core of the framework is a personal

submission agent that performs all submission stages [8] and watches over the efficient execution of the job. Adaptation to changing conditions is achieved by dynamic rescheduling. Once the job is initially allocated, it is rescheduled when performance slowdown or remote failure are detected, and periodically at each discovering interval. Application performance is evaluated periodically at each monitoring interval. The submission agent consists of the following components:

- Request Manager (RM): To handle client requests.
- Dispatch Manager (DM): To perform job scheduling.
- Submission Manager (SM): To perform the stages of job execution, including job migration.
- Execution Manager (EM): To execute each job stage.
- Performance Monitor (PM): To evaluate the job performance.

The flexibility of the framework is guaranteed by a well-defined API for each submission agent component. Moreover, the framework has been designed to be modular to allow adaptability, extensibility and improvement of its capabilities. The following modules can be set on a per job basis:

- Resource Selector (RS): Used by the Dispatch Manager to select the most adequate host to run each job according to the host's rank, architecture and other parameters.
- Middleware Access Driver (MAD): Used by the Execution Manager to submit, monitor and control each job stage.
- Performance Evaluator (PE): Used by the Performance Monitor to check the progress of the job.
- Prolog (P): Used by the Submission Manager to prepare the remote machine and transfer the executable, input and restart (in case of migration) files.
- Wrapper (W): Used by the Submission Manager to run the executable file and capture its exit code.
- Epilog (E): Used by the Submission Manager to transfer back output or restart (in case of stop) files and clean up the remote machine.

Therefore, RS interfaces Grid Information services (e.g. Globus pre-WS and WS MDS), MAD interfaces Resource Management services (e.g. Globus pre-WS and WS GRAM), Prolog and Epilog interfaces Data Management services (e.g. Globus GridFTP, Reliable File Transfer and Data Replication Service), Wrapper interfaces Execution services and PE interfaces Performance services. The result is that the Grid*Way* core is independent of the underlying middleware.

4.1 The Request Manager and Dispatch Manager

The client application uses the Grid*Way* client API or the DRMAA API [12] to communicate with the Request Manager in order to submit the job along with its configuration file, or job template, which contains all the necessary parameters for its execution. Once submitted, the client may also request control operations to the request manager, such as job stop/resume, kill or reschedule.

The Dispatch Manager periodically wakes up at each scheduling interval, and tries to submit pending and rescheduled jobs to Grid resources. It invokes the execution of the Resource Selector module, which returns a prioritized list of candidate hosts. The Dispatch Manager submits pending jobs by invoking a Submission Manager, and also decides if the migration of rescheduled jobs is worthwhile or not. If this is the case, the Dispatch Manager triggers a migration event along with the new selected resource to the Submission Manager, which manages the job migration.

4.2 The Submission Manager and Performance Monitor

The Submission Manager is responsible for the execution of the job during its lifetime, i.e. until it is done or stopped. It is invoked by the Dispatch Manager along with a selected host to submit a job, and is also responsible for performing job migration to a new resource. The Globus management components and protocols are used to support all these actions. The Submission Manager performs the following tasks:

- Prologing: Submission of Prolog executable.
- Submitting: Submission of Wrapper executable, monitoring its correct execution, updating the submission states and waiting for events from the Dispatch Manager.
- Cancelling: Cancellation of the submitted job if a migration, stop or kill event is received by the Submission Manager.
- Epiloging: Submission of Epilog executable.

This way, Grid*W*ay doesn't rely on the underlying middleware to perform preparation and finalization tasks. Moreover, since both Prolog and Epilog are submitted to the front-end node of a cluster and Wrapper is submitted to a compute node, Grid*W*ay doesn't require any middleware installation nor network connectivity in the compute nodes. This is one of the main advantages of the "end-to-end" architecture of Grid*W*ay.

The Performance Monitor periodically wakes up at each monitoring interval. It requests rescheduling actions to detect better resources when performance slowdown is detected and at each discovering interval.

4.3 The Execution Manager

In order to provide an abstraction with the resource management middleware layer, the Execution Manager uses a Middleware Access Driver (MAD) module to submit, monitor and control the execution of the Prolog, Wrapper and Epilog modules. The MAD module provides basic operations with the resource management middleware, like submitting, polling or cancelling jobs, and receives asynchronous notifications about the state of each submitted job. The use of standard input/output makes easy the debugging process of new MADs.

Currently, the are two MADs available. One, written in C, interfaces pre-WS GRAM services and other, written in Java, interfaces WS GRAM services. Java Virtual Machine (JVM) initialization time doesn't affect, since the JVM is initiated before the start of measurements.

5 Experiences

5.1 Application

In this work we have used the NGB Embarrassingly Distributed (ED) benchmark [13]. The ED benchmark represents the important class of Grid applications called Parameter Sweep Applications (PSA), which constitute multiple independent runs of the same program with different input parameters. This kind of computations appears in many scientific fields like Biology, Pharmacy, or Computational Fluid Dynamics. In spite of the relatively simple structure of these applications, its efficient execution on Grids involves challenging issues [14].

The ED benchmark comprises the execution of several independent tasks. Each one consists in the execution of the SP flow solver [15] with a different initialization parameter for the flow field. In the present work, we have used the FORTRAN serial version of the SP flow solver code. We have used a problem size of class A but, instead of submitting 9 tasks, as NGB class A specifies, we submitted more tasks in order to have a real high-throughput application.

For these experiments we have used a simple Resource Selector consisting of a list of resources, along with their characteristics (including the MAD that should be used to access each of them). Resources are used in a round-robin fashion, as long as they have free slots.

5.2 Testbed

In this section, we show the coordinated use of a research testbed with WS GRAM (described in Table 1) and a production testbed (described in Table 2), which is composed of some spanish sites enroled in EGEE, with pre-WS GRAM as part of the LCG (LHC Computing Grid) middleware. The whole testbed is connected by the Spanish National Research and Education Network (RedIRIS).

The resulting environment is highly dynamic and heterogeneous, due to the shared use of compute and network resources, the different DRMS, processors and network links, the different middleware and services, etc. In this case, we have submitted an array jobs with 100 tasks. We have imposed the limitation to only use four nodes simultaneously on each compute resource. In the following experiments, cygnus is used as client.

Table 1. Characteristics of the resources in the research testbed

Name	Site	Location	Nodes	Processor	Speed	Memory per node	DRMS
cygnus	UCM	Madrid	1	Intel P4	2.5GHz	512MB	-
ursa	UCM	Madrid	1	Intel P4	3.2GHz	512MB	fork
draco	UCM	Madrid	1	Intel P4	3.2GHz	512MB	fork
hydrus	UCM	Madrid	4	Intel P4	3.2GHz	512MB	PBS
aquila	UCM	Madrid	2	Intel PIII	600MHz	250MB	SGE

Table 2. Characteristics of the resources in the production testbed

Name	Site	Location	Nodes	Processor	Speed	Memory per node	DRMS
egeece	IFCA	Cantabria	28	2×Intel PIII	1.2GHz	512MB	PBS
lcg2ce	IFIC	Valencia	117	AMD Athlon	1.2GHz	512MB	PBS
lcg-ce	CESGA	Galicia	72	Intel P4	2.5GHz	1GB	PBS
ce00	INTA-CAB	Madrid	4	Intel P4	2.8GHz	512MB	PBS
ce01	PIC	Cataluña	65	Intel P4	3.4GHz	512MB	PBS

There are several differences between the version of Globus included in LCG and a Globus version installed out of the box. For example, the automatic generation of Grid map files, the use of GLUE schema for MDS, the use of BDII instead of GIIS, and the fact th at file systems are not shared by default between cluster nodes. In a previous work [16], we have shown the coordinated use of two Grid infrastructures, one based on Globus pre-WS services and one based on the LCG middleware, by only using Globus pre-WS protocols and interfaces. In this work, we have extended the modularity of the Grid*W*ay framework to the resource management interfacing layer, through the MAD, in order to support the coordinated use of both pre-WS and WS Grid services.

5.3 Results

Figures 3 and 4 respectively show the dynamic throughput achieved and scheduling performed during the four experiments. Experiment 1 reaches the maximum throughput (212 jobs/hour) since all resources are available. During experiment



Fig. 3. Dynamic throughput in the four experiments

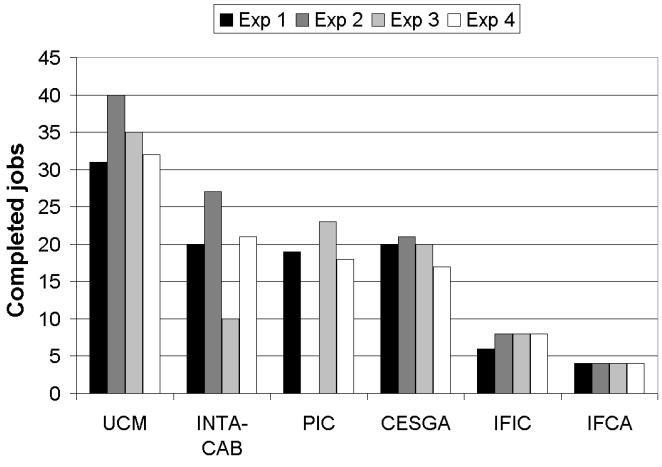


Fig. 4. Scheduling performed in the four experiments

2, PIC is unavailable, so no job is allocated to this site and the other sites receive more jobs. Therefore, the throughput drops considerably (154 jobs/hour). During experiment 3, INTA-CAB is partially busy, being only two nodes available for execution. This is reflected in the schedule (INTA-CAB receives half as jobs as in the first experiment) and in the achieved throughput (181 jobs/hour). During experiment 3, CESGA and PIC receive some Grid jobs not related to the experiment. In all the experiments, UCM receives more jobs than the other sites since it presents more compute nodes (10 vs. 4) due to the limitation of four simultaneously running jobs on the same resource.

In most experiments, throughput drops at the end (last five jobs). This is due to bad scheduling decisions (remember the simple RS used) or unexpected conditions triggering job migrations. If there are lots jobs and the testbed is saturated, their effects are hidden, but when few jobs remain, they arise.

6 Conclusions

We have shown that our proposed user-level Grid middleware, Grid*W*ay, can work over different Grid infrastructures and service technologies in a *loosely-coupled* way. In this case, we have shown the use of Grid*W*ay over a research testbed based on Globus WS Grid services and a production testbed based on Globus pre-WS Grid services, as part of the LCG middleware, demonstrating that Grid*W*ay can simultaneously work with both pre-WS and WS GRAM. The smooth process of integration of two so different infrastructures and services demonstrates that the Grid*W*ay approach, based on a modular, decentralized and "end-to-end" architecture, is appropriate for the Grid.

We would like to acknowledge all the institutions that have contributed resources to perform the experiments.

References

1. Foster, I., Kesselman, C.: Globus: A Metacomputing Infrastructure Toolkit. J. Supercomputer Applications **11** (1997) 115–128
2. Czajkowski, K., Ferguson, D.F., Foster, I., et al.: The WS-Resource Framework Version 1.0. Technical report (2004) Available at http://www.globus.org/wsrf.
3. Allan, R.J., Gordon, J., McNab, A., Newhouse, S., Parker, M.: Building Overlapping Grids. Technical report, University of Cambridge (2003)
4. Snelling, D., van den Berghe, S., von Laszewski, G., et al.: A UNICORE Globus Interoperability Layer. Computing and Informatics (2002) 399–411
5. Foster, I.: What Is the Grid? A Three Point Checklist. GRIDtoday **1** (2002) Available at http://www.gridtoday.com/02/0722/100136.html.
6. Foster, I., Czajkowski, K., et al.: Modeling and Managing State in Distributed Systems: The Role of OGSI and WSRF. Proc. IEEE **93** (2005) 604–612
7. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Technical report, OGSI Working Group – GGF (2002)
8. Schopf, J.M.: Ten Actions when Superscheduling. Technical Report GFD-I.4, Scheduling Working Group – GGF (2001)
9. Czajkowski, K., Foster, I., Karonis, N., et al.: A Resource Management Architecture for Metacomputing Systems. In: Proc. IPPS/SPDP Workshop on Job Scheduling Strategies for Parallel Processing. Volume 1459 of LNCS. (1998) 62–82
10. Graham, S., Niblett, P., et al.: Publish-Subscribe Notification for Web Services Version 1.0. Technical report (2004) Available at http://www.globus.org/wsrf.
11. Raicu, I.: A Performance Study of the Globus Toolkit and Grid Services via DiPerF, an Automated DIstributed PERformance Testing Framework. Master's thesis, University of Chicago, Computer Science Department (2005)
12. Rajic, H., Brobst, R., et al.: Distributed Resource Management Application API Specification 1.0. Technical report, DRMAA Working Group – GGF (2003)
13. Frumkin, M.A., Van der Wijngaart, R.F.: NAS Grid Benchmarks: A Tool for Grid Space Exploration. J. Cluster Computing **5** (2002) 247–255
14. Huedo, E., Montero, R.S., Llorente, I.M.: Experiences on Adaptive Grid Scheduling of Parameter Sweep Applications. In: Proc. 12th Euromicro Conf. Parallel, Distributed and Network-based Processing (PDP), IEEE CS (2004) 28–33
15. Bailey, D.H., Barszcz, E., Barton, J.T.: The NAS Parallel Benchmarks. J. Supercomputer Applications **5** (1991) 63–73
16. Vázquez, J.L., Huedo, E., Montero, R.S., Llorente, I.M.: Execution of a Bioinformatics Application in a Joint IRISGrid/EGEE Testbed. In: Proc. PPAM Workshop on Large Scale Computations on Grids (LaSCoG). LNCS (2005) (to appear).

Web-Services Based Modelling/Optimisation for Engineering Design

Ali Shaikh Ali[1], Omer F. Rana[1], Ian Parmee[2], Johnson Abraham[2], and Mark Shackelford[2]

[1] School of Computer Science and Welsh eScience Center, Cardiff University, UK
[2] CEMS, University of West of England, UK
Ali.ShaikhAli@cs.cardiff.ac.uk

Abstract. Results from the DIstributed Problem SOlving (DIPSO) project are reported, which involves the implementation of a Grid-enabled Problem Solving Environment (PSE) to support conceptual design. This is a particularly important phase in engineering design, often resulting in significant savings in costs and effort at subsequent stages of design and development. Allowing a designer to explore the potential design space provides significant benefit in channelling the constraints of the problem domain into a suitable preliminary design. To achieve this, the PSE will enable the coupling of various computational components from different "Centers of Excellence". A Web Services-based implementation is discussed. The system will support clients who have extensive knowledge of their design domain but little expertise in state-of-the-art search, exploration and optimisation techniques.

1 Introduction

A key theme in Grid computing is the capability to share services (representing different kinds of expertise) distributed across multiple institutions, and generally referred to as a "Virtual Organisation" (VO). The Distributed Problem Solving (DIPSO) project is based on extending such a model of a VO to integrate capability provided through different "Center of Excellence" (CoE). Each CoE in this instance provides very specific expertise – and ways in which such expertise can be accessed, but does not indicate precise details of how such an expertise is to be physically implemented within the center. Such centers may be geographically remote and several centers may offer similar services giving the client the option to select and link varying service providers using criteria relating to cost and reputation. Based on this model, we envision the availability of a "Modelling" Center, which given a data set can construct a model of this data set. Which technique is used to construct the model is not revealed to the user, only that different types of modelling capability is available, which varies in the accuracy of the model that will be built. A user may decide to construct a simple model (as it require less computation time or is less costly) in the first instance, and use this as a basis for decision making. Alternatively, a different

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 244–253, 2005.
© Springer-Verlag Berlin Heidelberg 2005

user may only be interested in a very precise model. This ability to not explicitly reveal the details of the modelling algorithm is useful to: (1) manage intellectual property of the owners of the modelling center, (2) enable updated or new modelling algorithms to be made available, (3) not require the user to know details about the types of modelling algorithms being supported. We describe a Problem Solving Environment that couples multiple CoEs to support the conceptual design phase, and illustrate the approach based on a Web Services-based implementation.

In the first instance, we assume also that each center only provides one kind of expertise. Hence, each center only supports one service. A Modeller and an Interrogator/Optimiser service are provided across two CoEs, supporting a range of search, exploration and optimisation techniques. The Interrogator service extracts information relating to design space characteristics either from data-based models generated within the Modeller, or directly from a parametric model that resides with a client. The availability of Web Service standards (such as WSDL, SOAP), and their widespread adoption, including by the Grid community as part of the Web Services Resource Framework (WSRF), indicates that exposing the Modeller and Interrogator as Web Services is likely to be useful to a significant user community. Exposing the capabilities as Web Services also enable these to be combined with other third party services, allowing the Modeller and Interrogator to be embedded within existing applications. The remainder of this paper is structured as follows. An overview of related research is provided in section 2. The DIPSO framework is discussed in section 3. Implementation of the framework as Web Services follows in section 4. Finally, performance results based on the use of Web Service technologies are discussed in section 7.

2 Related Work

The availability of distributed resource integration lends itself well to design optimisation – especially the capability to couple multiple expertise, and the use of high performance resources to undertake parametric search. Nimrod-G [3] is the most commonly utilised software for parametric design search, and has been used in a variety of applications. It can, however, only be applied to one aspect of the DIPSO approach mentioned above, and may be used to configure and setup a multi-parameter search. No particular search algorithm is specified as part of this work. Our approach can therefore we easily integrated with Nimrod-G. The GeoDise [2] project focuses on engineering design, and is most closely aligned with our approach. The focus in the GeoDise project has been the capability to set up multiple design experiments – which may be executed concurrently – to evaluate parameter ranges of interest to a design. Such design parameters are then sent to Computational Fluid Dynamics (CFD) programs, for instance, to evaluate their usefulness in a simulation. Our focus has been primarily on conceptual design (the early stages of the design process), where a designer needs to consider various available options prior to moving to a more detailed design via specialist packages (such as CFD solvers).

3 Framework Overview

The DIPSO framework consists of two services: (1) The Modeller service, (2) the Interrogator/Optimisation service. The Modeller comprises a number of ancillary data processing techniques plus neural network and statistical modelling software for the generation of models from incoming data sets passed into the system by the client. The Interrogator/Optimiser extracts information relating to design space characteristics either from data models generated within the Modeller or directly from a parametric model that resides with a client [4].

The Interrogator prototype comprises a number of space-sampling techniques to identify well-distributed points within the design space, and standard hill-climbers which search from these points. A clustering algorithm applied to the hill-climber output can provide an indication of the number and distribution of local optima to the client. The clustering output is passed through a simple rule-set which determines which, if any, further optimisation technique should be applied to the problem. The Optimiser currently comprises three search and optimization algorithms: a genetic algorithm (GA), a simulated annealing algorithm (SA) and a tabu search algorithm (TS). The Knowledge Repository contains information relating to problem characteristics and solutions from the Interrogator/Optimiser. The architecture is presented in figure 1. The Modeller would be the property of one center of expertise whereas the Interrogator/Optimiser would be owned by another.

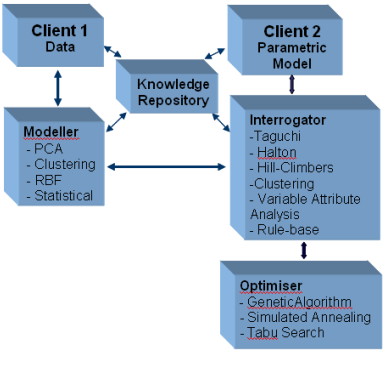


Fig. 1. Initial Architecture

4 Web Services to Support Modeller/Interrogator

The Modeller and Interrogator/Optimiser have been implemented as Web Services. Each provides a generic set of operations that abstract a range of different internal algorithms. A single call to a Modeller service, for instance, may lead to aggregation of results from a number of different components. The exact workflow between these components is not revealed through the Modeller interface.

4.1 Interrogator Web Service

When dealing with a parametric model as opposed to a statistical or RBF model generated by the Modeller the interrogator Web Service extracts information relating to design space characteristics directly from the parametric model Web service that resides with a client. The client is not required to pass the parametric model to the interrogator Web service but the URL address of the model (a support tool to enable a client to convert the model into a Web Services is also provided). A parametric model, in this instance, is an executable binary – which may be run with different sets of input parameters. The Web service has the following main operations: `setupInterrogator(URI model, int dimension, int minMax)`: prepares for an interrogation task. The client passes the URI address of the model, the number of dimensions in the model and a flag indicating whether he wants to minimize or maximize the solution; `runQuickInsection()`: provides a means to get a representative distribution of inspection points; `runInDepthInspection(URI model, int dimension, int minMax)`: provides a more detailed view via a greater number of inspection points; `runHillClimb()`: runs a hill-climber on each of the sample points and the variable set and best result of each hill-climber are then passed to the client, and `runCluster()`: runs a near-neighbours clustering algorithm on the sample points. The clustering results are passed through a rule-base which determines which stochastic search algorithm is most appropriate.

Similarly, the parametric model Web Service is an interface to a client's model. The interface provides two main operations: (1) `getFitness(double[][] sample points)`: returns the objective value of the passed sample points, and (2) `getRange()`: which returns the parameter range associated with the parametric model.

Two Web services are provided for plotting the results of the Interrogator:`ChartPlotter` which generates two dimensional graphs, such as bar, line and area charts, scatter plots and bubble charts, amongst others. ChartPlotter Web service is an interface implementation for JFreeChart. The main operation of this service is the `plot2D(File data point, String chartType, String chartTitle, String xAxisLabel, String yAxisLabel)`. The Web Service returns the plotted chart as a Portable Network Graphics (PNG) image. The second Web Service is `3DPlotter` which may be used to produce a scatter plot, histogram and a box plot. 3DPlotter is an interface implementation for JMathPlot. The main operation of this service is `plot3D(File data points, String plotType)`. The Web Service returns the plotted graph as a PNG image.

5 Framework Architecture

The framework architecture is shown in Figure 2. In the first instance, a user must find the address of the Web portal which interacts with the interrogator and the parametric model Web Services. The Web portal implements a proforma in Java Server Pages (JSP) which provides the user with guidance as a set of sequential steps. Via the proforma, the user is asked about the type of model

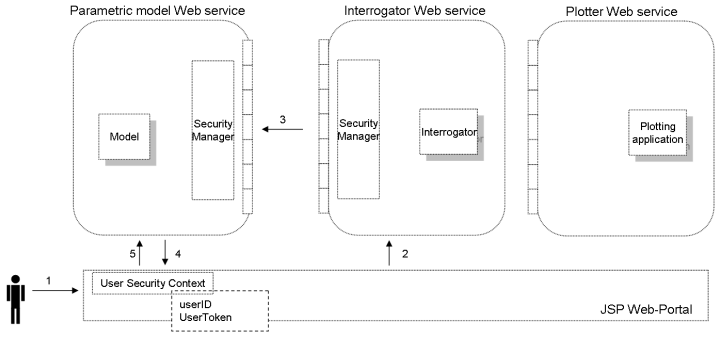


Fig. 2. Web Service Architecture

they have (such as the number of input parameters, a reference to the model Web Service, etc). The proforma is intended to provide a guided process by which a user can identify optimal solutions for their particular parametric model, subject to constraints of execution time. The sequence of steps are discussed in details in section 6.

The Web-portal makes use of a Security Manager to constrain the access to the parametric model itself, or the results from the Optimiser. The mechanism is an implementation of Security Assertion Markup Langauge (SAML) standard – which provides a mechanism for making authentication and authorization assertions, and a mechanism for data transfer between different cooperating domains without the need for those domains to lose the ownership of that information. SAML specifies three main components that are implemented in our framework: (1) assertions, (2) protocol, and (3) binding. We make use of three assertions: `authentication` – which validates a users identity, `attribute` – contains specific information about a user, and `authorization` – identifies what the user is authorized to do. The assertions are provided by a SAML authority, which is implemented as a component within the Web-portal. For the protocol and binding components we use OpenSAML [1] which is a set of open-source libraries in Java that are used to build, transport, and parse SAML messages. The following scenario (illustrated in figure 2) demonstrates how we use SAML in our framework:

1. A user supplies security credentials to the Web portal – which includes an authentication token for accessing the user's parametric model. On receiving the security credential, the Web-Portal creates a security context (SAML assertion) for the user and stores it on the server and produce a SAML artifact of this assertion.
2. On accessing the interrogator Web service, the portal provides the interrogator with the user's SAML artifact.
3. The user may chose an operation provided by the interrogator Web Service that requires access to the parametric model. In this situation, the interrogator sends the user's SAML artifact to the parametric model, which sends a SAML request to the Web portal.

4. The Web portal responds with a message containing the SAML user assertion. The assertion includes the token supplied by the user (step 1) for accessing the parametric model. Subsequently, at the parametric model site, the assertion is processed and a decision for accessing the parametric model is either granted or denied.

6 Interactions

Interaction with the system is supported through the Web portal that automatically interconnects the different Web Services together. This portal makes use of a workflow graph that is hidden from a user. A user is presented with a set of pre-defined questions to ascertain what the user wants. The sequence of events is illustrated in Figure 3.

On initialisation, the portal displays a setup page to the user, providing online guidance and possible options. Via the setup page, the user is asked to enter the parameters that describe the client's model. The parameters include: the number of dimensions in the model, whether the client wants to minimize or maximize the solution, and the URL address of the parametric model.

Subsequently, the portal invokes the `setup(int dimensions, int MinMax, String URL)` operation of the interrogator Web Service – which checks the va-

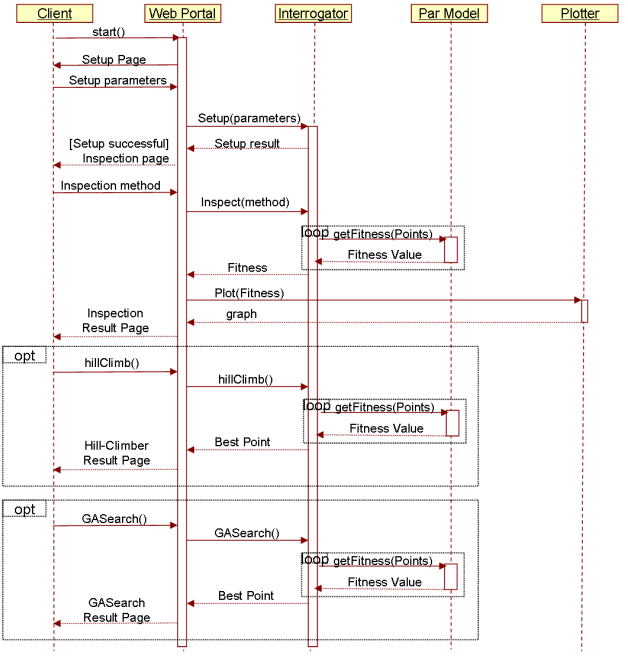


Fig. 3. Web Service Architecture

lidity of the parameters, establishes a connection with the parametric model and initializes an instance of the interrogator component. The Web service returns the result of the setup operation. If either the parameters are invalid, or the connection with the parametric model cannot be established, an error is sent to the user via the portal.

If the setup was successful, the portal provides the user with the inspection page providing the following options: (1) Quick Inspection: this will provide a good spread of inspection points; (2) In Depth: this will take a more detailed view via a greater number of inspection points but will take longer to process. On choosing one of the inspection methods, the `inspect(String method)` operation is called. The second option relates to the introduction of Halton sequences [5] which provide a far denser sampling of the space than the first option, Taguchi [6], but would be more expensive computationally. The resulting sample solutions (i.e. variable sets) are then passed to the client's parametric model which returns the calculated objective value for each solution.

A complete list of the solution variable sets and their objective values is then returned to the client. This is achieved by calling the `getFitness` operation. Figure 4 shows a screen shot of the result. The client then has three options: (1) If some or all of the calculated solution objective values are erroneous then abandon the process, review and modify the model and re-present for further sampling and testing; (2) if all the objective values appear sensible and one or more prove to satisfy the client's requirements in terms of a sufficiently high-performance solution then accept these solutions and terminate the process; (3) continue and find better solutions.

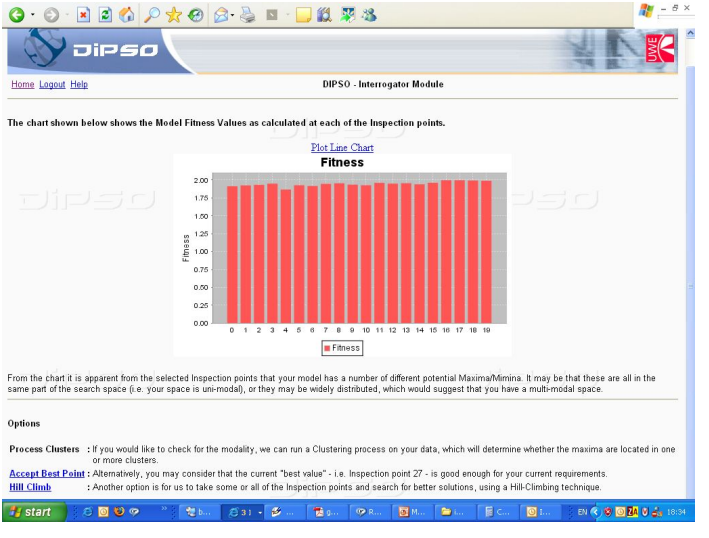


Fig. 4. A screen shot of the result that the client get on running quick inspection on a parametric model

If a user wishes to continue, a Simplex hill climber can be started from: (1) the most fit solution points, (2) the best 10% of the solution points, or (3) all solution points. This is achieved by calling the `runHillClimb` operation on the interrogator Web Service. If option (3) is chosen, and if all the hill climbers converge to very similar solutions in terms of their variable and objective function values then the client is advised that it is probable that little further improvement is possible i.e. it is likely that the solution space is unimodal / monotonic and the optimal solution has been identified.

If the client still wishes to continue search for possible better solutions then a near-neighbours clustering algorithm [7] is introduced to the set of 'best' hill-climber solutions – by invoking the `runCluster` operation on the Interrogator Web Service. Additional rules can be used to determine further optimisation. For instance: (1) If just one cluster of solutions is identified but the Euclidean distance between each solution in the cluster is significant then either a simulated annealing or tabu search algorithm is initiated within the cluster region i.e. it is assumed that this is a region containing a number of local optima; (2) if more than one cluster is identified, and they are in diverse parts of the overall design space, then it is considered likely that the overall search space is multi-modal i.e. many local optima may exist across the space. It is then assumed that a more global search process is required and a genetic algorithm is introduced. In either case the best 10% of all solutions identified by the optimization are returned to the client.

7 Experiments

The primary goal of our experiments is to evaluate the performance of our framework, and better understand interactions between the interrogator Web service and the Web-enabled parametric model. We undertake two experiments. The first evaluates the performance of distributing the interrogator and model at two different sites – thereby modelling a user interacting with a Center. The second experiment evaluates the performance of using SOAP for multiple invocations within the same setup. In both experiment we will evaluate the interaction between the interrogator and a Web-enabled parametric model. The parametric model we use is provided by Systems Engineering and Assessment (SEA) Ltd – an industry partner in the project. The parametric model is for a remote-operated undersea vehicle. Design variables within this conceptual whole-system model include overall vehicle dimensions, characteristics of single or hybrid power sources, fuel and equipment loadings, etc – 19 in total. The interrogator and the SEA parametric model Web services are deployed using Axis and Tomcat and installed on a Windows XP platform. The interrogator Web service is hosted at UWE in Bristol while the SEA parametric model Web service is hosted at Cardiff.

Experiment 1: In this experiment the performance of distributing the interrogator and the parametric model is evaluated (and compared with having them at the same site). The inspection operation (used by the interrogator) makes

use of an established space sampling technique (based on a Halton sequence). The resulting sample solutions (i.e. variable sets) are then passed to the client's parametric model which returns the calculated objective value for each solution. The later process is time consuming, as each time the interrogator passes the variable sets to the parametric model, it invokes the `getFitness(double[][] sample points)` operation on the remotely located parametric model. Figure 5 presents a comparison between the time required to compute the objective value for 40 solutions. There is a 50% loss in performance as a result of distribution (with this difference remaining uniform over all 40 solutions). The distributed `getFitness` operation ran for 52.20s while the local configuration completed in 38.01s.

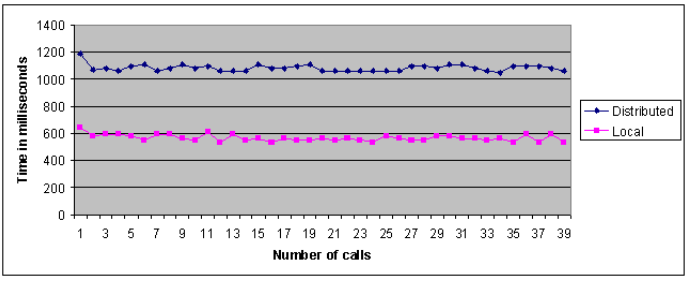


Fig. 5. Running the `getFitness` operation – local and distributed calls

Experiment 2: We evaluate the performance of using the SOAP protocol and standard TCP sockets to send the sample points over the network. Figure 6 shows a comparison between SOAP and TCP for 40 invocations. Our results indicate that the performance of the two is comparable. We therefore do not see any performance loss with the use of the XML encoding often reported for SOAP. One possible reason for this is the small message size used in experiments, leading to equivalent delays.

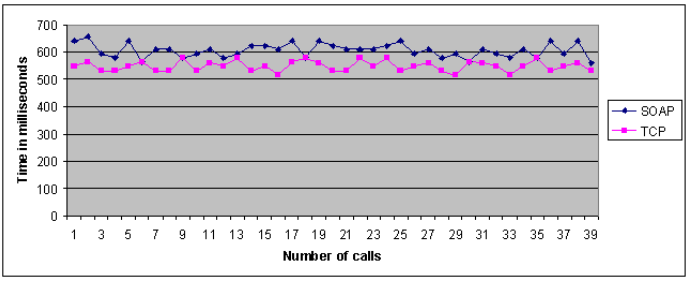


Fig. 6. A comparison between running the *getFitness* operation using SOAP vs. TCP

8 Conclusion

A system to support distributed modelling and design is presented, which makes use of Web Services to integrate a variety of different functionality. The approach makes use of a Center Of Excellent approach, whereby modelling and optimisation expertise can be coupled. Performance issues with the approach are also discussed.

References

1. "OpenSAML Project", available at: `http://www.opensaml.org/`. Last Viewed: June 2005.
2. G. Xue, W. Song, S. J. Cox, and A. J. Keane, "Numerical Optimisation as Grid Services for Engineering Design", Journal of Grid Computing, Vol. 2 No. 3, 2004, pp 223-238. Project Web site at: `http://www.geodise.org/`. Last Visited: June 2005.
3. R. Buyya, D. Abramson, and J. Giddy, "Nimrod-G Resource Broker for Service-Oriented Grid Computing", IEEE Distributed Systems Online, in Vol. 2 No. 7, November 2001.
4. I.C. Parmee, J. Abraham, M. Shackelford, D. Spilling, O. F. Rana, A. Shaikhali, "Introducing Grid-based, Semi-autonomous Evolutionary Design Systems", International Conference on Engineering Design (ICED 05), Melbourne, August, 2005.
5. L. Kocis and W. Whiten, "Computational Investigations of Low-Discrepancy Sequences", ACM Transactions on Mathematical Software, Vol. 23, No. 2, 1997, pp 266-294.
6. G. Taguchi, "Systems of Experimental Design", Kraus International Publications, 1987.
7. R. A. Jarvis and E. A. Patrick, "Clustering using a Similarity Measure Based on Shared Near-neighbours", IEEE Transactions on Computers 22[11], 1973.

Workflow Management System Based on Service Oriented Components for Grid Applications*

Ju-Ho Choi, Yong-Won Kwon, So-Hyun Ryu, and Chang-Sung Jeong**

Department of Electronics and Computer Engineering,
Korea University,
Anamdong 5-1 Sungbukku,
Seoul 136-701, Korea
Tel: +82-2-3290-3229; Fax: +82-2-926-7620
{jhchoi, luco, messias}@snoopy.korea.ac.kr
csjeong@charlie.korea.ac.kr

Abstract. To efficiently develop and deploy parallel and distributed Grid applications, we have developed the Workflow based grId portal for problem Solving Environment(WISE). However, the Grid technology has become standardized related to Web-services for Service-oriented architecture and the workflow engine of WISE is insufficient to meet the requirements of a service-oriented environment. Therefore, we present a new workflow management system, called the Workflow management system based on Service-oriented components for Grid Applications(WSGA). It provides an efficient execution of programs for computational intensive problems using advanced patterns, dynamic resource allocation, pattern oriented resource allocation, and configures Web-service as an activity of workflow in Grid computing environment. In this paper, we propose the WSGA architecture design based on service-oriented components, and functions for using Web-services and increasing system performance. Also, we show an implementation method, and report the system performance evaluation of the WSGA architecture design.

1 Introduction

Grid Computing[1] is an efficient technology for taking advantage of heterogeneously distributed computing resources. The user doesn't need to know how to use acquired distributed resources. Grid computing is the fundamental infrastructure for E-Science, requiring very high performance computation, very large-scale data management, composition tools, collaborative environments and knowledge-based services. In a powerful Problem Solving Environment(PSE), a user can easily utilize the grid environment for solving problems, workflow

* This work has been supported by a Korea University Grant, KIPA-Information Technology Research Center, University research program by Ministry of Information & Communication, and Brain Korea 21 projects in 2005.

** Corresponding author.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 254–263, 2005.
© Springer-Verlag Berlin Heidelberg 2005

technology is very important because it can coordinate various applications on multiple distributed resources.

Recently, a Grid computing environment appears to be changing to a Service Orient Environment(SOE). Web-services[2][3] are the realized model of a Service Oriented Architecture(SOA)[4] and Grid technology is presented in the Open Grid Service Architecture(OGSA)[5], using Web-service technology. The Globus Toolkit 3(GT3)[6] actualizes OGSA by Web-service based components for Grid services. Therefore, a Grid workflow management system has to work based on OGSA.

Our previous workflow management system, WISE[7] was implemented based on Globus Toolkit 2. It provides Grid portal service using graphic user interface and advanced workflow patterns for Grid applications. But, it does not support Grid environment based on OGSA, execution of sub-workflow for scalability, dynamic resource allocation, pattern oriented resource allocation, and use of Web-services. Therefore, we can not use a lot of Grid resources and Web-services efficiently.

In this paper, we present a new advanced architecture of workflow management system, which enables Web-service to be used as an activity of workflow, and provides functions such as dynamic resource allocation, pattern oriented resource allocation and group workflow for increasing performance and scalability.

The outline of our paper is as follows: In section 2, we describe previous workflow systems. In Section 3, we analyze our requirements. In Section 4, we describe the components of WSGA architecture. In Section 5, we explain the pattern oriented resource allocation method of WSGA. In Section 6, we experiment with computational intensive application and analyze results, In Section 7, we present our implementation method using the GT3. Finally, a conclusion will be given in Section 8.

2 Related Work

2.1 Previous Grid Workflow Systems

The Pegasus[8] is designed to concrete an abstract workflow onto the Grid environment. This system converts logical data to physical data such as the mapping of a logical file name into an executable file. The Grid Services Flow Language(GSFL)[9] is XML-based and allows the specification of workflow descriptions for Grid services in the OGSA framework. It has been defined using XML schemas. It is not a workflow system but a workflow definition language, enabling definition of Web-service interfaces. The JOpera[10] is a service composition tool offering a visual composition language, differentiated from others by using visual syntax to describe data flow, linking the input data and output data of two tasks and control flow by connecting tasks. These workflow systems are insufficient in supporting Grid environment based on OGSA and advanced patterns with dynamic resource allocation. Therefore, we need a new Grid workflow management system to solve these problems.

2.2 Grid Portal System, WISE

The workflow based Grid portal system, WISE has a 3-tier architecture, consisting of clients, web application server, and a network of computing resources. The web application server in the middle tier is augmented with Grid-enabling software to provide accesses to Grid services and resources. The system is designed as a multi-layered structure, exploiting the Model-View-Controller(MVC) design pattern and Commodity Grid(CoG)[11] technology. The application engine in the web server controls and organizes the overall portal through proper execution of application logic components, according to the client's request, which in turn carry out Grid services through the Grid service interface, activating the presentation module and generating application specific display to be transmitted to the client's browser. WISE provides advanced workflow patterns which enable a user to make and execute parallel programs easily. This is an advantage and different point as compared with other workflow systems.

3 Requirements

The goal is service-oriented components based design and performance improvement. To achieve this, the workflow engine of WISE needs a new architecture based on OGSA. The engine must be changed to use a service oriented Grid environment, and be added components, which use Web-services. We have to improve functions of WISE engine about workflow control, such as sub-workflow, dynamic resource allocation, pattern oriented resource allocation for increasing performance and scalability. It provides not only solutions of computational intensive problems, but also interoperability between different platforms or services. We approach our goal using GT3, providing Grid services based on Open Grid Service Infrastructure(OGSI) and Web-service technology.

4 System Architecture

4.1 Overall Architecture

We show our workflow management system architecture in Figure 1. The WSGA is a workflow engine of WISE. The WSGA consists of the Workflow Manager, Activity Manager, Web Service Binding Manager, Data Conversion Manager, and Resource Information Manager. We will explain these functions in detail, in the following sections.

4.2 Workflow Manager

The Workflow Manager processes a workflow description, sub-workflow, and schedules jobs to execute. The Workflow Manager consists of the Workflow Parser, RSL Maker, Workflow Scheduler, and Executable Dispatcher. The Workflow Parser parses Grid Workflow Description Language(GWDL) which is an XML-based workflow description, used to specify workflow of WSGA. The RSL

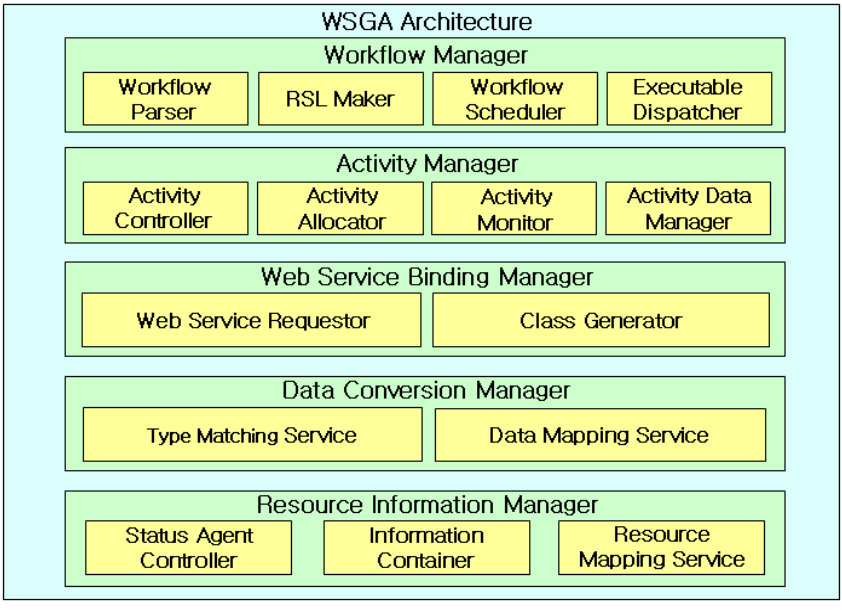


Fig. 1. WSGA Overall Architecture

maker generates an RSL file from GWDL to submit and allocate jobs in distributed resources. The RSL Maker supports both RSL in GT2 and XML based RSL in GT3 for using various Grid resources. The Workflow Scheduler has order of jobs to execute from a GWDL file. The Workflow Scheduler sends information of activity execution to the Activity Manager, and receives the state of activities to run next jobs. The user can specify sub-workflow, which is called Group Workflow in this system for increasing scalability. It is described in GWDL by the user. Each of resources has partial functions of WSGA such as the Workflow Manager, Activity Manager and Data Conversion Manager like a light workflow engine. The Workflow Parser in a parent node reads a GWDL file and sends a portion of Group Workflow to child nodes. The user can assign a particular node as a child node, otherwise a child node is allocated by the Resource Mapping Service of Resource Information Manager. The child node parses a Group Workflow description by using the Workflow Parser, independently. Activities are allocated and processed by the Workflow Scheduler and Activity Manager in a child node. The child node sends notification to the Workflow Scheduler of parent node when the Group Workflow was processed. Therefore, we can obtain scalability for solving computational intensive problems. The Executable Dispatcher enables transferring a execution file when a workflow has started.

4.3 Activity Manager

The Activity Manager allocates jobs to resources according to the activity types. The Activity Manager consists of the Activity Controller, Activity Allocator, Activity Monitor, and Activity Data Manager. The Activity Controller gets activity

type and calls appropriate Allocator, Monitor and Data Manager. The Activity has two groups according to the Grid environment and connection type between workflow engine and activity. Each of groups has two types.

***According to the Grid Environment*:** GT2 and GT3. GT2 type activity uses functions of GT2, which are not based on service-oriented components, but the GT3 type activity uses functions of GT3, which are based on an OGSA using WSDL. Activity Controller selects proper functions for allocating job and transferring data.

***According to the Connection type*:** Loosely Coupled Type(LCT), Tightly-Coupled Type(TCT). LCT only executes files without notifying state in detail. TCT notifies state in detail and provides socket connection for peer-to-peer communication between TCT activities. It can use data conversion and mapping services easier than LCT. TCT is faster than LCT to communicate with each other, but legacy programs are difficult to use TCT. Therefore, we provide two types of activity to improve convenience and performance for various environments.

The Activity Allocator allocates jobs to resources and the Activity Data Manager moves data from a source site to a destination site. The Activity Monitor observes and gathers activity state.

4.4 Web Service Binding Manager

For interoperability between diverse services, workflow management system has to support the Web-service. The Web Service Binding Manager consists of the Web Service Requestor and Class Generator. The user inputs an address of Web-service, and the Web Service Requestor gets WSDL files using Web-service URL. The Class Generator generates Java package which consists of class files. A client program is made by the user using class files in WISE portal. A WISE portal provides convenient user interfaces to write client program for using Web-services. A complete client program is treated and allocated as an activity. Thus, we can configure a workflow using Web-services. Type Matching Service and Data Mapping Service can use, if necessary. Therefore, Web-services are configured and used in a workflow. We show a workflow configuration using WSGA in Figure 2.

4.5 Data Conversion Manager

The Data Conversion Manager consists of the Type Matching Service and Data Mapping Service. Data conversion is required between activities, or an activity and a Web-service. Type Matching Service provides a proper data type or content. For example, we provide image file conversion service which changes data type from jpeg to bitmap for running experimental application program using Web-service. Data Mapping Service parses a document and arranges data. It sends proper data to next nodes through socket connection. It is used for TCT. For instance, when a TCT activity received a notification from Web-services, Data Mapping Service mapped a notification to integer for notifying to TCT activity because TCT activity uses representation of events as an integer in WSGA.

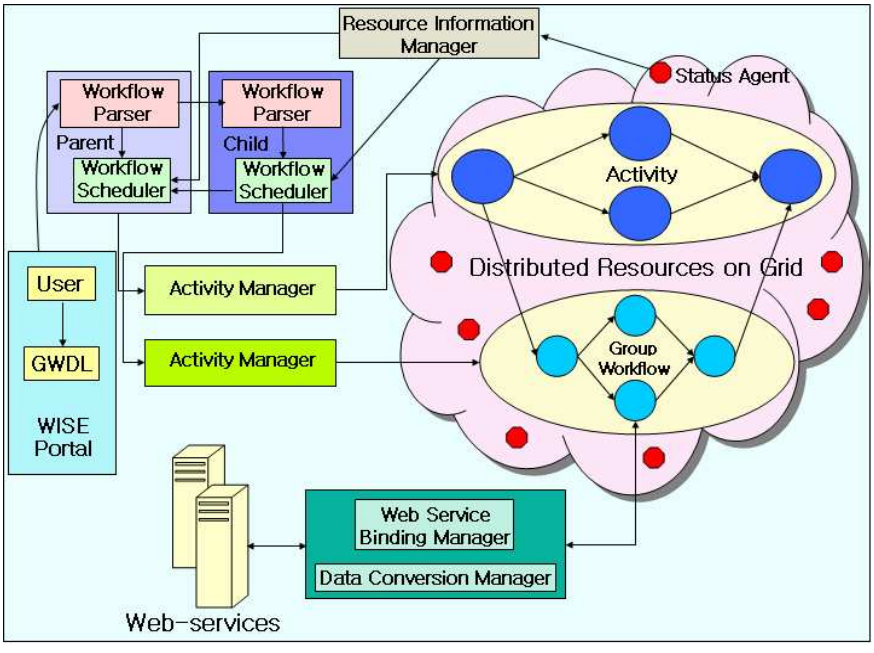


Fig. 2. Workflow Configuration using WSGA

4.6 Resource Information Manager

The Resource Information Manager gathers status of resources and reports the best resource to Workflow Scheduler at that time for dynamic resource allocation which is needed for increasing system performance. The Resource Information Manager consists of the Status Agent Controller, Information Container, and Resource Mapping Service. The Status Agent Controller creates and controls Status Agents. Status Agent is allocated to each of resources. It gathers the following types of data from resource: CPU clock, average CPU load, memory statistics, OS type, network type, network status, number of processes and logical disk volumes. Network status counts number of error packets for sending and receiving data. Memory statistics included free memory size and total memory size. Dynamic information such as average CPU load, network status, number of processes and memory statistics are obtained per one minute. The Information Container gathers data from Status Agents. The Resource Mapping Service provides a proper resource when the workflow scheduler requests. It can allocate appropriate resources following priority: average CPU load, network status and number of processes. Selected resources already meet primary conditions such as resource state is not failure, free memory size, and logical disk volume. If a user does not choose a host name or condition for an activity, the Resource Mapping Service allocates proper resources, automatically.

5 Pattern Oriented Resource Allocation

WISE provides advanced patterns to describe various parallelism because the other workflow patterns are too simple to describe the structure of parallel and distributed application on Grid environments. Grid applications need large data transfer between activities in peer-to-peer communication. However, communication overhead for transferring data is an main factor to drop system performance. To prevent unnecessary data transfer, not only workflow model should allow user to describe data flow explicitly but also resource allocation should be optimized. Therefore, we integrated both advanced control patterns and data flow into a pattern with an optimized resource allocation method because distributed Grid applications need various control parallelism and parallel data communication. We show various patterns with optimized resource allocation method and classify them in three groups: sequential flow, parallel flow, and mixed flow.

***Sequential flows*:** Sequence, XOR-Split, XOR-Join, and Loop. A sequential pattern represents a sequential control flow between two activity nodes. XOR-Split is a conditional choice. Only one activity can be executed with a predicate. XOR-Join is like XOR-Split, excepts reverse flow. Loop is like a "while" statement of C program language. We do not allow job migration using dynamic resource allocation to other resource except fault occurrence because communication process for transferring data to a new resource spends too much time.

***Parallel flows*:** AND-Split, AND-Join, AND-Loop. They are implemented for using simple parallel fork and synchronization. AND-Split executes all the next nodes. Each of nodes has its own thread. AND-Join waits until all nodes are completed. AND-Loop is a variant of simple loop construction that AND-Split follows each iteration. We allow dynamic resource allocation for using parallel or pipeline algorithms except AND-Loop. In these flows, a previous resource is reused like one of the post resources and transfers data to other resources using multicasting algorithm. It can prevent wasted time for transferring data to a new resource.

***Mixed flows*:** Multi-Split and Multi-Join. Multi-Split allows us to choose multiple activities among the output activities, and selected activities are executed simultaneously. Multi-Join is like XOR-Join but it allows multiple choice. We can allow dynamic resource allocation like parallel flows in the same way. A previous resource is reused and uses multicasting algorithm for transferring data to reduce data transfer time.

6 Experiment

We experiment 3D volume rendering on WSGA using pattern oriented resource allocation, dynamic resource allocation and Group workflow for comparing with our previous workflow system, WISE. Volume rendering is a powerful tool for visualization of 3D object and requires highly computational costs. Among the technique developed for volume rendering, ray casting is considered the most

Table 1. Machine Specifications and Relative Performance for 3D Volume Rendering

Machine type	M_1	M_2
Model	Pentium 4 PC	Pentium 4 PC
CPU	Pentium 4	Pentium 4
Clock(GHz)	1.7	2.4
Memory(MBytes)	1024	1024
OS	Linux (Redhat 9)	Linux (Redhat 9)
running time(sec)	102.011	74.142
relative perf.	1.000	1.376

Table 2. Experiment Results

number of machines		1(M_2)	2($M_{2,2}$)	4($M_{2,2,2,2}$)	8($M_{1,1,1,1,2,2,2,2}$)
expected speedup		1.0	2.0	4.0	6.907
WSGA	time (sec)	74.142	41.778	21.501	12.824
	speedup	1.0	1.775	3.448	5.782
WISE	time (sec)	72.582	43.234	23.455	14.710
	speedup	1.0	1.679	3.095	4.934
Improvement efficiency(%)		-2.149	3.368	8.331	12.821

simple and well suited for parallel processing. In ray casting, a ray is passed through each pixel of the screen from the view point, and the volume data along the ray are sampled, accumulated to provide the final color and opacity of the corresponding pixel. We implemented 3D volume rendering application for WSGA. WSGA distributed data set to nodes and executed the workflow to start volume rendering. We use 8 node of Pentium 4 PCs, connected by 100Mbps Ethernet. A size of data set is 256 x 256 x 256, and image screen has 1024 x 1024 pixels. The detailed information of hardware and software are shown in Table 1.

Since two type machines have different computing power, we have measured the relative performance with M_1 as reference machine for comparing the execution time of the identical speed up. The expected speed up is computed as a sum of each relative performance of participating machines. The relative performance of the machines obtained by executing the identical sequential 3D volume rendering program. In Table 2, we shows the execution time, speed up and efficiency on WSGA, comparison with WISE, according to the number of machines. The efficiency represents improvement of speed up, comparison with WISE. Only one machine is used, WISE is faster than WSGA because WSGA consists of service oriented components. However, as the number of machines increases, WSGA shows relatively faster and follows identical speed up than WISE due to dynamic resource allocation, pattern oriented resource allocation and group workflow.

7 Implementation

The Globus Toolkit is a software toolkit, used to program grid-based applications. We implemented WSGA in Java with the GT3 library. The CoG Kits allow Grid users, Grid application developers, and Grid administrators to use, program, and administer Grids from a higher-level framework. The CoG Kit

is used in GT3 and provides important functionality. We implemented WSGA using the Java API, which further uses Grid services including the GridFTP, RFT, WS-GRAM, and Information Service. Activity Manager implemented using GT2 and GT3. GT2 type activity uses GridFTP for transferring files and the GRAM for allocating jobs to resources. GT3 type activity uses RFT service for transferring files and the WS-GRAM for allocating jobs to resources. RFT is a service, used to interact between GridFTPs using XML-based description. Web Service Binding Manager is implemented using Apache Axis, a necessary API for GT3. Resource Information Manager is implemented by the Web Service Information Service in GT3. We can obtain simple or detailed resource information. A Java-based provider, called Simple Provider by GT3, includes simple resource information. It produces XML output in the form of a Java output stream. The Status Agent sends a data of resource in detail to Simple Provider because Simple Provider does not investigate detailed status of resource such as average cpu load, network status and number of processes. The Service Data Container in GT3 gathers information from Simple Providers and responds with a host name when the Resource Mapping Service is requested. The WSGA was implemented using loosely coupled components. Therefore, the components can adapt easily when a new environment or architecture is designed.

8 Conclusion

In this paper, we have presented the Workflow management system based on Service-oriented components for Grid Applications(WSGA), providing workflow management functions based on GT3. The Grid computing environment appears to be changing towards a SOE, consisting of loosely coupled service components. The GT3 provides an OGSA environment using Web-service technology. Thus, we implemented WSGA using the GT3 to support a SOE. This environment has the advantage of rapid adaptation to new architectures and environments. The WSGA uses not only GT2 based Grid resources but also GT3 based Grid resources, called Grid services. Also, WSGA can configure Web-service in a workflow and improved dynamic resource allocation method related to advanced patterns and group workflow for increasing system performance and scalability. In future work, we are extending an activity or a workflow which can provide WSDL using GT4, which is a new version of Globus Toolkit. We will deploy each activity or workflow as a service provider. Therefore, we will develop the WSGA to provide Web-services for solving computational intensive problems based on Grid environment.

References

1. I. Foster, C. Kesselman, S. Tuecke. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations.", International J. Supercomputer Applications, 2001.
2. W3C. Web Services, http://www.w3.org/2002/ws/.

3. Web Services Description Language (WSDL) Version 2.0 Part 0: Primer, http://www.w3.org/TR/2004/WD-wsdl20-primer-20041221.
4. SOA and Web Services, http://java.sun.com/developer/.
5. I. Foster, C. Kesselman, J. Nick, S. Tuecke, Open Grid Service Infrastructure WG, Global Grid Forum. "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration." June 22, 2002.
6. Globus Toolkit 3, http://www-unix.globus.org/toolkit/.
7. Y.W. Kwon, S.H. Ryu, J.S. Park and C.S. Jeong, "A Workflow-Based Grid Portal for Problem Solving Environment", NPC 2004, Wuhan, China, October 18-20, 2004.
8. Ewa Deelman, James Blythe, Yolanda Gil, Carl Kesselman, Gaurang Mehta, Sonal Patil, Mei-Hui Su, karan Vahi, Miron Livny, Pegasus: Mapping Scientific Workflows onto the Grid, Across Grid Conference 2004, Nicosia, Cyprus.
9. Sriram Krishnan, Patrick Wagstrom, and Gregor von Laszewski. "GSFL: A Workflow Framework for Grid Services". Argonne National Laboratory, Preprint ANL/MCS-P980-0802, Aug 2002.
10. Cesare Pautasso, Gustavo Alonso. "JOpera: a Toolkit for Efficient Visual Composition of Web Services", International Journal of Electronic Commerce (IJEC), 9(2):107-141, Winter 2004/2005.
11. CoG Kits, http://www.cogkit.org.

Life Science Grid Middleware in a More Dynamic Environment

Milena Radenkovic and Bartosz Wietrzyk

School of Computer Science and IT, University of Nottingham,
Jubilee Campus, Wollaton Road, Nottingham, NG8 1BB, UK
{mvr, bzw}@cs.nott.ac.uk

Abstract. This paper proposes a model for integrating a higher level Semantic Grid Middleware with Web Service Resource Framework (WSRF) that extends the prototype presented in [1] informed by issues that were identified in our early experiments with the prototype. WSRF defines generic and open framework for modeling and accessing stateful resources using Web Services and Web Service Notification standardizing publish/subscribe notification for Web Services. In particular we focus on using WSRF to support data integration, workflow enactment and notification management in the leading EPSRC e-Science pilot project. We report on our experience from the implementation of our proposed model and argue that our model converges with peer-to-peer technology in a promising way forward towards enabling Semantic Grid Middleware in mobile ad-hoc networks environments.

1 Introduction

The Grid and Web Services technologies are currently converging and many Grid middleware projects are in transition from pre-Web Service versions (e.g. Globus Toolkit 2 [2], LCG2 [3]) to the new ones based on Web Services. The two major European Grid initiatives, namely UK's National Grid Service (NGS) [4, 5] and EU's Enabling Grids for e-Science (EGEE) [6] have chosen them as the interface for services they provide. Although there is a widespread agreement within the Grid community about the long-term wisdom of adopting a Service Oriented Architecture (SOA) and Web Service technologies as the basis for building a robust and stable Grid infrastructure, there is a short-term problem with such an approach. SOA is still very much work in progress. There are currently many proposed specifications but very few mature standards, which have gained general acceptance and are supported with robust tooling. While standardized and widely adopted specifications are essential for the production deployment, it is also important for the communities to explore the emerging standards and estimate their value before they are adopted in the production environment [7]. Stable and promising Web Services Resource Framework (WSRF) [8, 9] implementations are being evaluated by the National Grid Service (NGS). There is a commitment from NGS that WSRF implementations will be deployed and tested on the NGS production level service [4, 5].

Web Service technology is a major step from the human-centric Web, providing communication between a human and an application, to the application-centric Web

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 264 – 273, 2005.
© Springer-Verlag Berlin Heidelberg 2005

providing communication between applications. Web Services are platform and programming language independent which is extremely important for integrating heterogonous systems and development of loosely coupled distributed systems. They do not only standardize the message exchanges but also the descriptions of interfaces and service discovery [10]. Web Services in their basic form are already widely accepted and adopted by both commercial and academic institutions. The Web Service standards are used in many ongoing e-Science projects providing either Grid or Semantic Web middleware, they are even considered to be 'an actualization of the Semantic Web vision' [11].

Web Services from their definition are stateless, i.e. implement message exchange with no access or use of information not contained in the input reference [12]. Even though it is sufficient for many applications, there are many which require the notion of state for their communication. In these cases, the lack of any standards for expressing the state forces designers to use custom solutions, which are highly incompatible between particular systems, making their interoperability and integration increasingly difficult.

WSRF standardizes the design patterns and message exchanges for expressing state, i.e. data values that persist across, and evolve because of, Web Service interactions [12]. WSRF is considered an 'instruction set of the Grid' [13]. Besides WSRF there is another important standard defined. It is Web Service Notification (WSN) [14] which covers publish/subscribe notification for Web Services.

In this paper, we focus on these two standards that we believe are very important for the future of Web Services, reporting on development of our prototype work initially introduced in [1]. In order to investigate how they can support an existing, complex middleware project, we took myGrid as our example. MyGrid [15] is one of the leading EPSRC e-Science pilot projects that focus on the development of the open source Semantic Grid middleware for Bioinformatics supporting *in silico* experiments. MyGrid is building high-level services for data and application integration like resource discovery, distributed query processing and workflow enactment. Additional services are provided to support scientific method such as provenance management, change notification and personalization. As this project will finish shortly a proper exit strategy is essential. We believe that integration with WSRF and WSN are a very important part of it, increasing myGrid's interoperability and taking advantage of deployed services and resources, in particular provided by NGS. We also want to improve myGrid's scalability by providing a more distributed architecture.

In this paper we propose a novel model for integrating myGrid with WSRF/WSN. We report on our experience from the integration we made discussing its advantages and challenges. We envisage that our approach will converge with the peer-to-peer technology paving the way of Semantic Grid Middleware into mobile ad-hoc network (MANET) environment. The novelty of our approach comes from using peer-to-peer technology not only to provide self-organization and scalability but also to increase the reliability of storage and message delivery [16].

This paper is organized as follows. Section 2 presents an overview of myGrid architecture and proposes a model for integrating myGrid with WSRF and WSN. Section 3 evaluates our model, reports on the experiences form the integration we made and presents our peer-to-peer approach for utilizing mobile ad-hoc environments. Section 4 gives conclusions.

2 Model for Integrating MyGrid with WSRF/WSN Standards

This section defines our novel model for integrating myGrid with WSRF and WSN standards. We begin with describing the current myGrid's architecture. MyGrid is an EPSRC e-Science pilot project aiming at development of Open Source Semantic Grid middleware for Bioinformatics supporting *in silico* experiments. MyGrid is building high-level services for data and application integration like resource discovery, distributed query processing and workflow enactment. Additional services are provided to support scientific method such as provenance management, change notification and personalization [15].

In myGrid the *in silico* experiments are depicted as workflows comprising an interconnected set of inputs, outputs and processors, which are either local routines or Web Services [17]. The enactment of workflows, which are XML documents, is performed by the enactment engine. All the data including not only input, output and workflows but also connections between them and particular people together with the structure of involved institutions is stored in the myGrid Information Repository (MIR). All the entities within myGrid have their own unique IDs – Life Science ID (LSID) issued by the central authority. The real time inter-component event-driven notification is provided by the myGrid Notification Service (MNS) [18]. MyGrid adapts the service-oriented approach i.e. not only helps to integrate, possibly third party, Web Services, but all its components have Web Service interfaces.

Its key components, relevant for modeling with WS-Resources due to having notion of state include MIR entities, workflow enactments, and enactment services. MyGrid notification infrastructure can be adapted to the WS-Notification [14] specification. All the WS-Resource Qualified Endpoint References (WSRQER) of myGrid's WS-Resources will comprise the address of the appropriate Web Service and LSID.

2.1 MyGrid Data

Currently all MIR entities have types. Every type is associated with an XML schema of an XML document describing attributes of an entity. Entities are stored in the relational database, and accessed through the Web Service interface using document call format. The supported operations are presented in Table 1. We map the MIR entities on WS-Resources. Every single entity becomes a WS-Resource, every entity type - a resource type and every entity's attribute - a WS-Resourve's property. We use the XML schema for MIR information model as a source of types for resource properties documents for WSRF. The only change involves replacing the attributes containing LSIDs as references to other entities with their complete WSRQERs. The mapping between the current and WSRF operations are shown in Table 2. The WSRF data exchanges are simpler than their current equivalents, because the WS-Resource type is designated by the URL part of the WSRQER specified in the SOAP header and the LSID is passed as the ResoureProperty of the WSRQER.

As it can be clearly spotted, Table 1 contains two more entries than Table 2. The StoreEntity operation was omitted because WSRF does not provide any standardized message exchanges for creating WS-Resources. According to the WS-Resource Factory pattern [12] we keep the operations StoreEntity and GetEntityCollection in

Table 1. Operations currently supported by MIR

Name	Input	Output	Description
StoreEntity	XML document with all entity properties; entity type	Generated LSID	Stores a new entity into MIR and generates an LSID for it.
GetEntity	Sequence of one or more pairs of entity type and an LSID	Sequence of XML documents with all entity properties	Retrieves properties of documents of given LSIDs.
GetEntity Collection	Entity type; optionally a sequence of restrictions comprising an attribute name, a condition and a value	Sequence of entities' LSIDs	Retrieves a sequence of LSIDs for entities of a given type, according to given criteria (similar to the SQL *where* clause).
DeleteEntity	Sequence of one or more pairs of an entity type and an LSID	Status	Deletes one or more entity element from MIR.
UpdateEntity	Sequence of pairs of an XML document describing entity's properties and an LSID; entity type	Status	Updates properties of one or more entities of a given type (not all types are updatable).

Table 2. Mapping between current and new WSRF operations

Current operation	WSRF operation	Inputs	Outputs	Standard	Comments
GetEntity	GetResource Property	Property name	Property value	WS-Resource Properties	It is possible to specify properties to query.
	GetResource Properties	Sequence of property names	Sequence of property values		
Delete Entity	Delete	None	None	WS-Resource Lifetime	We can allow either immediate or delayed destruction.
	SetTermination Time	Termination time	New termination time and current time		
Update Entity	SetResourceProperties/UpdateResourceProperties	Sequence of pairs comprising a property name and a new value	None	WS-Resource Properties	We do not have to provide all attributes to update only a subset of them.

their current form in the Resource Factory and Discovery (RFAD) service, only changing their names to Create and Query and their output from LSIDs to WSRQERs. RFAD is responsible for creating and discovering WS-Resources and knows where all data resources are hosted. This approach allows every single resource to be stored on a different machine. In a large deployment, there could be a few public RFAD services running on specialized machines and several machines hosting WS-Resources having its own RFAD service, only for the use by public RFAD services, see Figure 1.

In our model, every WS-Resource replacing a MIR entity provides NotificationProducer interface as describe in WS-ResourceProperties [19] and WS-ResourceLifetime [20]. The notification process would be performed in a standard pattern as described in WS-BaseNotification [21], providing notifications about destruction of WS-Resources and changes of their properties.

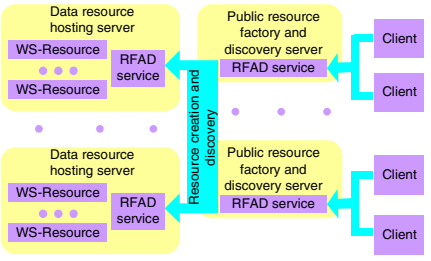


Fig. 1. Proposed data resources architecture

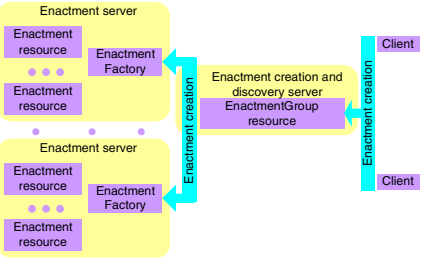


Fig. 2. Proposed workflow enactment architecture

2.2 Workflow Enactment

In myGrid workflows are depicted as XML documents written in SCUFL, which can be enacted by the Freefluo enactment engine. An important part of this process is collection of provenance, which comprises intermediary results and technical metadata including type of processors, status, start and end time and descriptions of the performed operations. This information can be useful in investigating how the erroneous or unexpected results have been achieved [17].

In the current release of myGrid Freefluo is integrated with Taverna and must be run locally from it, what is inconvenient for a user in case of long lasting enactments. It is planned to decouple them, by providing Freefluo as a Web Service hosted on a remote machine and manually putting its URL into the Taverna's configuration file. Using only one enactment server for a deployment can cause a bottleneck. Even if we use more, the current approach does not support any load balancing, because a user chooses one explicitly.

In our model shown in Figure 2 every enactment is virtualized as a WS-Resource (Enactment resource), which could be running on any of the available enactment servers. Its properties include: StartTime (the time stamp of the resource creation), Operation (reference to the Operation data resource [22] containing input parameters and workflow description in the SCUFL language), Status (statuses of the processors' invocations and intermediate results) and Topics (notification topics including only one topic ResourceTermination for enactment finish, error or cancellation). Enactment resource provides following operations: GetResourcePoropoerty and GetResourceProperties from WS-ResourceProperties for acquiring the resource's properties, Destroy from WS-ResourceLifetime for the immediate cancellation of the enactment, Subscribe from WS-BaseNotification for subscribing to notifications about the finish, error or cancelling of the enactment.

The EnactmentGroup resource acts as a WS-ResourceGroup [23] of all Enactment resources to allow their seamless discovery. It provides one custom operation Create, which will create an Enactment resource, taking as a parameter a WSRQER to the Operation data resource [22], referencing all the input data. It returns a WSRQER to the newly created Enactment resource. The EnactmentGroup selects an enactment server, on which the Enactment resource should be created, providing some load balancing. Internally the resource creation is done by invoking Enactment Factory

Web Service on the enactment server. Client can query at any time the status of the resource using WSRQER either from the Create operation or WS-ResourceGroup's Entry Properties. When the Enactment resource is terminated, either by the successful or erroneous end of the enactment or a client's cancellation, the OperationInstance data resource [22] is created, which holds all the provenance data.

2.3 Notification Infrastructure

In our model, any notification producer can manage its own subscriptions, so in a simple deployment no notification service would be necessary. However to provide more scalability notification brokers can be introduced. In WS-Notification sense, a broker implements both notification consumer and producer interfaces, so it can decouple notification consumer and producer forming even a very complex structure, where one broker receives notifications form another and passes it further [24]. The broker is subscribed to the subset of notification producer's topics and exposes them as his property. Depending on the particular deployment, a broker can have one of two objectives: aggregating topics managed by different WS-Resources to allow their seamless discovery or distributing the task of message delivery to increase its speed and decrease network congestion.

The role and the interface of a notification broker in WSRF is very different from the current myGrid Notification Service [18]. On one hand notification brokers are much more lightweight not supporting complex features like quality of service negotiation; on the other hand they are more generic allowing building complex structures.

3 Discussion

3.1 Overview of the WSRF/WSN Toolkits for MyGrid

We choose Apache WSRF (formerly Apollo) [25] for WSRF and Pubscribe (formerly Hermes) [26] for WSN both developed by The Apache Software Foundation. We choose Pubscribe because it is built on top of Apache WSRF and Apache WSRF as it is the only one offering all together Java API, dynamic creation of WS-Resources, callbacks for modification of WS-Resources and free, Open Source status. That is described in more detail in [1].

3.2 Evaluation of the Model

Our model changes the centralized myGrid Information Repository (MIR) into a set of cross-referenced WS-Resources, which can be hosted on different machines possibly administrated by different institutions. We introduce a distributed and scalable enactment infrastructure supporting load balancing. That all provides more flexible architecture, which can be easily extended after the system is deployed, so it can seamlessly grow with the increasing number of users, fully transparently to them.

It is possible because in WSRF WS-Resources are addressed by WS-Resource Qualified Endpoint References (WSQERs) containing URLs of Web Services

describing where a Web Service part of a WS-Resource is located, so it is possible to host them at different network locations, in a transparent to a user way. Moving existing WS-Resources will be even easier, when the WS-RenewableReferences specification [9] becomes available. It will allow seamless updating WSQERs when they become outdated.

In our model, the notification infrastructure is more scalable, distributed and lightweight than originally in myGrid. The notification brokers can be used to form any topology either to aggregate topics published by various WS-Resources or to distribute the process of delivering messages. As every WS-Resource can manage its own subscribers, the notification brokers are optional and not required for simple deployments. Because the notification infrastructure is compliant with WSN, it is compatible with the third-party infrastructure for delivering messages and notification clients that are available now or in the future.

Introduction of WSRF and WSN provides one coherent and logical interface for operating on user's data, workflow enactment and notification infrastructure. That decreases the design effort needed to integrate various myGrid components or in future to integrate myGrid with third party tools, and UK's National Grid Infrastructure [4].

As the data model of WS-Resources is declared in WSDL descriptions of their Web Services, it becomes explicit for the clients. It makes evolution of the data model easier and the service consumers can even automatically adapt to its changes. It allows the gradual modification of the data model to fulfill changing users' requirements in fully transparent way even after deployment.

The WS-Resources implemented using Apache WSRF [25] and Pubscribe [26] have the form of servlets, which can be deployed to any servlet container like for example Jakarta Tomcat [27]. Their deployment comprises installing third party products (Java 2 Standard Edition, Jakarta Tomcat or an alternative servlet container, MySQL and LSID Launchpad), deployment of WAR files and modification of appropriate configuration files. Therefore the integration with WSRF/WSN does not make the deployment of myGrid [28] more demanding.

3.3 Experience from the Integration

Working on the implementation of our proposal we identified following challenges. To take the full advantage of WSRF the LSID references must be replaced with WSQER. That means making major changes in both the existing data model and the code. Using Apache WSRF, which currently is the only WSRF toolkit that fulfilled our requirements demands a lot of coding effort. It is mainly due to the extensive plumbing between the XMLbeans used to handle properties and a means of persistence, which in case of myGrid is the mySQL database supported with Hibernate [29]. There is also an important performance issue when using Apache WSRF or Java WS Core from GT4 to access data stored in a database. When a resource is created or accessed it becomes a Java object residing in the server's memory, until the service is shut down or the object is destroyed i.e. permanently removed. In case of huge databases it means a very inefficient use of the system's memory.

3.4 Towards Self-organizing Grids

Currently myGrid is meant to be deployed on the fixed network infrastructure. Such an infrastructure is not always available for a scientist doing for example a field research. Deploying myGrid on ad-hoc, possibly mobile, networks would mean facing some challenges. Firstly, the naming scheme depends on the DNS infrastructure, which must be pre-configured, preventing self-organization. Secondly, the state of WS-Resources is available only when a machine hosting it is on-line, what heavily limits the reliability in mobile ad-hoc network (MANET) environments.

As our future research we are going to address these issues. We plan to alleviate the need of DNS by using Distributed Hash Tables (DHTs) basing on the approach suggested in [30]. We also plan to increase the reliability in MANET environments by providing distributed caching of the WS-Resource (WSR) state [16]. The state data will be available even when the WSR itself is off-line.

This approach will lead to more scalable and reliable data storage offering easy deployment with minimal administration effort, due to lack of centralization, cooperative caching and automatic configuration. Similar approach can be used to simplify the deployment and increase the reliability of remote access to scientific devices monitored or controlled over the network. Such an approach can be particularly useful in areas where the research takes place, which offer unreliable or none pre-planned network infrastructure.

We also consider using overlay networks and application level routing to increase the reliability of the notification delivery. The WSN approach is vulnerable to a

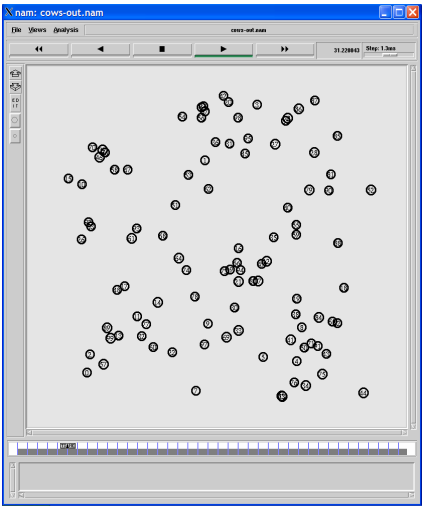


Fig. 3. Network simulation of 99 wireless Grid services and one user in 750m x 750m topology. User moves at the speed of 3-4 m/s starting from the bottom left corner and never stops. Services move at the speed of 1-4 m/s making stops for up to 20s.

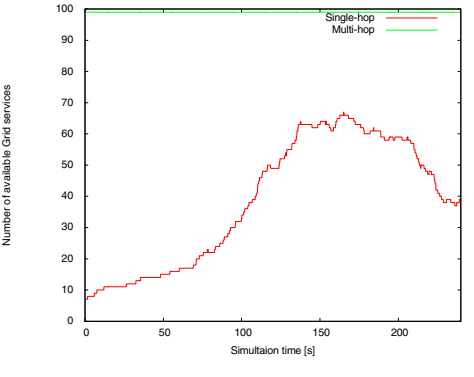


Fig. 4. Grid services available for the user – both within the radio range of her transceiver (single-hop) and over multiple hops

broker or notification producer going off-line. Using application level routing it is possible to distribute the responsibility for the message delivery among the forwarding nodes [16].

Currently to test our approach we are designing and developing a large scale simulation of application level DHTs using the *ns-2* network simulator [31] and the CanuMobiSim framework [32]. We want to examine the influence of dynamics of queries and the underlying network workload on the reliability of the storage and message delivery and performance of the queries. The example simulation is shown on Figure 3. Currently it comprises randomly generated movements of 99 Grid services and a user placed in a topology of 750m x 750m which in future will communicate using an application level DHT protocol. Services are wireless sensors placed on monitored animals. Figure 4 shows how many services were available for the user during the simulation. While the number of services within the range of user's wireless transceiver was limited, all the services were available in case of using multi-hop communication. That justifies using multi-hop peer-to-peer communication for such scenarios.

4 Conclusions

We proposed a novel model of integrating myGrid and WSRF that helps myGrid benefit in terms of scalability and interoperability. We also adapted the myGrid Notification Service, to the WSN specification. In this paper we define and evaluate our model and report on our experiences from the implementation of our proposal. Finally we argue that our approach will converge with the peer-to-peer technology to utilize mobile ad-hoc network environments. The myGrid data resources are already fully implemented and we are planning to implement the proposed workflow enactment architecture.

We proved that WSRF/WSN standards are well suited for the complex higher level middleware but applying these standards to the existing projects can lead to a significant coding effort. The approach presented here is universal and can be applied for standardization of other existing higher level middleware projects.

References

1. Wietrzyk, B., Radenkovic, M.: Semantic Life Science Middleware with Web Service Resource Framework. In: Proc. Fourth All Hands Meeting (2005)
2. Globus Toolkit 2.2.[Online]. Available: http://www.globus.org/gt2.2/
3. (2005) LHC Computing Grid Project (LCG) Home Page.[Online]. Available: http://lcg.web.cern.ch/LCG/
4. Geddes, N. GOSC and NGS Status.[Online]. Available: http://www.gridpp.ac.uk/gridpp12/GOSCNGS-status-Jan2005.ppt
5. Geddes, N., Richards, A. Grid Operations Support Centre (GOSC).[Online]. Available: http://www.nesc.ac.uk/talks/507/NESC_BBSRC_GOSC_241104.pdf
6. Loomis, C., Hahkala, J., Orellana, J. EGEE Middleware Architecture and Planning. EGEE [Online]. Available: https://edms.cern.ch/document/476451/
7. Atkinson, M., et al.: Web Service Grids: An Evolutionary Approach. OMII, (2004)

8. WSRF - The WS-Resource Framework.[Online]. Available: http://www.globus.org/wsrf/
9. Czajkowski, K., et al.: The WS-Resource Framework. Oasis, (2004)
10. Cerami, E.: Web Services Essentials. O'Reilly & Associates, Inc., Sebastopol, Canada (2002)
11. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American (2001)
12. Foster, I., et al.: Modeling Stateful Resources with Web Services. IBM, (2004)
13. Priol, T.: Objects, Components, Services for grid middleware: pros & cons. In: Proc. European Grid Conference (2005)
14. Graham, S., et al.: Publish-Subscribe Notification for Web services. IBM, (2004)
15. Stevens, R. D., Robinson, A. J., Goble, C. A.: myGrid: personalised bioinformatics on the information grid. Bioinformatics 19 (2003) i302-i304
16. Fall, K., et al.: Reliability in MANETs with Overlays. In: Proc. Peer-to-Peer Mobile Ad Hoc Networks - New Research Issues (2005)
17. Oinn, T., et al.: Taverna, lessons in creating a workflow environment for the life sciences. In: Proc. GGF10 (2004)
18. Krishna, A., Tan, V., Lawley, R., Miles, S., Moreau, L.: myGrid Notification Service. In: Proc. UK e-Science All Hands Meeting 2003 (2003)
19. Graham, S., Treadwell, J.: Web Services Resource Properties 1.2. Oasis, (2004)
20. Srinivasan, L., Banks, T.: Web Services Resource Lifetime 1.2. Oasis, (2004)
21. Graham, S., et al.: Web Services Base Notification. (2004)
22. Alpdemir, N., Ferris, J., Greenwood, M., Li, P., Sharman, N., Wroe, C.: The myGrid Information Model. University of Manchester, Design note Manchester (2004)
23. Maguire, T., Snelling, D.: Web Services Service Group 1.2. Oasis, (2004)
24. Graham, S., et al.: Web Services Brokered Notification. (2004)
25. Apache WSRF.[Online]. Available: http://ws.apache.org/wsrf/
26. Pubscribe.[Online]. Available: http://ws.apache.org/pubscribe/
27. (2005) The Jakarta Site - Apache Jakarta Tomcat.[Online]. Available: http://jakarta.apache.org/tomcat/
28. Nick Sharman, K. W., Tom Oinn, Kevin Glover, Nedim Alpdemir, Chris Wroe: The myGrid Installation Guide. (2005)
29. Bauer, C., King, G.: Hibernate in Action. Manning, Greenwich, USA (2004)
30. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A Scalable Content-Addressable Network. In: Proc. SIGCOMM (2001)
31. The Network Simulator - ns-2.[Online]. Available: http://www.isi.edu/nsnam/ns/
32. Stepanov, I. CANU Mobility Simulation Environment (CanuMobiSim).[Online]. Available: http://canu.informatik.uni-stuttgart.de/mobisim

A Grid-Aware Implementation for Providing Effective Feedback to On-Line Learning Groups

Santi Caballé[1], Claudi Paniagua[1], Fatos Xhafa[2], and Thanasis Daradoumis[1]

[1] Open University of Catalonia, Department of Information Sciences,
Av. Tibidabo, 39-43, 08035 Barcelona, Spain
{scaballe, cpaniagua, adaradoumis}@uoc.edu
[2] Dept. of Languages and Informatics Systems, Polytechnic University of Catalonia,
Jordi Girona Salgado 1-3, 08034 Barcelona, Spain
fatos@lsi.upc.es

Abstract. Constantly providing feedback to on-line learning teams is a challenging yet one of the latest and most attractive issues to influence learning experience in a positive manner. The possibility to enhance learning group's participation by means of providing appropriate feedback is rapidly gaining popularity due to its great impact on group performance and outcomes. Indeed, by storing parameters of interaction such as participation behaviour and giving constant feedback of these parameters to the group may influence group's motivation and emotional state resulting in an improvement of the collaboration. Furthermore, by feeding back to the group the results of tracking the interaction data may enhance the learners' and groups' problem solving abilities. In all cases, feedback implies constantly receiving information from the learners' actions stored in log files since the history information shown is continuously updated. Therefore, in order to provide learners with effective feedback, it is necessary to process large and considerably complex event log files from group activity in a constant manner, and thus it may require computational capacity beyond that of a single computer. To that end, in this paper we show how a Grid approach can considerably decrease the time of processing group activity log files and thus allow group learners to receive selected feedback even in real time. Our approach is based on the master-worker paradigm and is implemented using Globus technology running on the Planetlab platform. To test our application, we used event log files from the Basic Support for Collaborative Work (BSCW) system.

1 Introduction

Feedback in Computer-Supported Collaborative Learning (CSCL) environments [1] is recently receiving a lot of attention [3], [4], [5] due to its positive impact on the motivation, emotional state, and problem-solving abilities of groups in on-line collaborative learning [2], [4]. It aims to influence group participants in a positive manner by means of a steady tracking of parameters related to group functioning, task performance and scaffolding [6] and by giving a constant feedback of these parameters to the group. Therefore, when users participate in a collaborative learning experience, they may enhance their abilities by increasing their knowledge about

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 274 – 283, 2005.
© Springer-Verlag Berlin Heidelberg 2005

others in terms of cognitive processes and skills of the students and the group as a whole in solving problems, individual and group effectiveness regarding participation and interaction behavior, social support and help and so on.

The supply of efficient and transparent feedback to users in both synchronous and asynchronous modes is a significant challenge. Users are continuously interacting with the system (creating documents, reading others' contributions, etc.) thus generating a lot of events, which once collected, they are classified, processed, structured and analyzed [7]. As a consequence of the complex knowledge provided to participants (e.g., constant and automatic learner's assessment according to quantitative and qualitative parameters of the interaction) we need to capture all and each type of possible data that could result to a huge amount of information that is generated and gathered in data log files.

Our experience at the Open University of Catalonia [8] has shown the need to monitor and evaluate real, long-term, complex, collaborative problem-solving situations through data-intensive applications that provide efficient data access, management and analysis. To implement the collaborative learning activities and capture the group interaction we use the Basic Support for Cooperative Work (BSCW) [9] as a shared workspace system which enables collaboration over the Web by supporting document upload, group management and event service among others features. Despite BSCW's event service provides awareness information to allow users to coordinate their work, there is no support for feedback. Furthermore, BSCW provides neither log file processing nor tools for analyzing the processed information [2]. BSCW records the interaction as a huge amount of ill-structured information with a high degree of redundancy that requires an efficient data processing system to analize this complex information.

Therefore, there is a strong need for powerful solutions that record the large volume of interaction data and can be used to perform an efficient interaction analysis and knowledge extraction. In the literature, however, questions related to efficiently process the information obtained from group activity have been, to the best of our knowledge, hardly investigated. Some approaches [10], [11] consider the processing of the data just for a specific purpose, limiting thus both the specific provision of the knowledge extracted and the scope of the developed tools. In addition, they do not address the issue of processing time requirements that might result from the huge amount of data that are to be processed, which is a common issue in collaborative learning environments. The ultimate aim is to efficiently extract essential knowledge about the collaboration and to make it available to users as feedback.

Moreover, the need to make the analyzed information available in real time entails that we may come across with processing requirements beyond those of a single computer. Yet, the lack of sufficient computational resources is the main obstacle for processing large amounts of data log files in real time. In real situations this processing tends to be done later, after the completion of the learning activity, thus having less impact on it [2].

Recently, Grid technology is increasingly being used to reduce the overall, censored time in processing data by taking advantage of its large computing support. The concept of a computational Grid [12] has emerged as a way of capturing the vision of a networked computing system that provides broad access not only to massive information resources, but to massive computational resources as well. Thus, in this paper,

we show how a Grid-based application can be used to match the time processing requirements.

A preliminary study was conducted [2] to show that a Grid approach based on the Master-Worker (MW) paradigm [14] might increase the efficiency of processing a large amount of information from group activity log files. This allowed us to develop a real Grid-aware prototype that shows (i) how easily we are able to offload onto the grid the online processing of log data from the collaborative application, (ii) how a simple MW scheme suffices to achieve considerable speed-up, (iii) the gain provided by the Grid approach in terms of relative processing time and, (iv) the benefits of using the inherent parallel and scalable nature of Grid while the input log files are growing up in both number and large size.

The rest of the paper is organized as follows. In Section 2 we show the process of creating feedback and exemplify a real situation. Section 3 provides a Grid-based approach to present both a sequential and parallel processing of event log files to be used by Grid nodes whereby Section 4 shows the most representative computational results achieved. Finally, we conclude in Section 5 by drawing the most representative conclusions achieved and outlining ongoing work.

2 The Process of Providing Effective Feedback to On-Line Teams

Providing useful, effective, heterogeneous, yet structured feedback to collaborative learning activity is a complex process. As part of the process of embedding information and knowledge into CSCL applications [2], [7], this consists of four separate, indispensable stages: *collection of information, processing, analysis* and *presentation*. The entire process fails if one of these stages is omitted.

During the first stage, the most important issue while monitoring group activity is the efficient collection and storage of a large amount of event information generated by the high degree of interaction among the group participants. Given that such a large amount of informational data may need a long time to be processed, collaborative learning systems have to be designed in a way that classifies and pre-structures the resulting information effectively. The aim is, on the one hand, to correctly collect the group activity and, on the other hand, to increase the efficiency during the later data processing in terms of analysis techniques and interpretations. To that end, we propose a solution consisting of classifying information by means of three generic group activity categories [6], namely *collaborative learning product*, *group functioning* (i.e. individual and group effectiveness regarding participation and interaction behavior), and *scaffolding* (i.e. social support and task or group functioning help oriented services), which represent high-level collaborative learning processes.

Therefore, in order to constantly and immediately feed back all the information generated to on-line groups, once this information activity has been correctly collected and classified we may come across the issue of demanding computational requirements while processing this information [4]. In order to facilitate this step, CSCL applications may structure this information as log files in a way that takes advantage of the parallelism of a distributed environment such as Grid in order to process several files (e.g. all the groups in a classroom) at the same time and thus considerably reduce the overall computational time to process them [13]. As a result, it is possible

for these applications to process a large volume of collaboration activity data and make the extracted information available even in real time.

To that end, during the second stage, *processing*, we propose the following generic steps so as to correctly structure the event information for later processing and analysis [7]: by classifying the event information and turning it into persistent data, we store it in the system as structured files. These files contain all the information previously collected from the system log files in specified fields. Next, we predefine the structured files in accordance with certain criteria such as time and workspace, which characterize all group collaboration. Our goal is to achieve a high degree of granularity of log files. Thus, during later data processing, it is possible to concatenate several structured files so as to obtain the appropriate degree of granularity (e.g. all groups in a classroom for each 12 hours). This makes it possible to efficiently parallelize data processing depending on the characteristics of the computational resources.

During the *analysis* stage, the processed information is analyzed and interpreted in order to extract the appropriate knowledge according to the kind of feedback selected. The final stage, *presentation*, is to show the students the knowledge obtained, thus providing them appropriate feedback and metacognition.

At this point, in order to clarify the process of creating effective and useful feedback in a real situation, we briefly exemplify the case of enhancing the learners' motivation by constantly showing the current number of contributions in an asynchronous discussion process involving hundreds of groups and thousands of students.

For this example's purposes, we take the system's log file that collects all the current collaborative activity for a specific time (e.g. 15 minutes). This information is first classified according to learners, groups, time and type of action so that it is only related to the learner who generated events, the time when they occurred and the type of the action performed. Then, the given log file is partitioned into multiple log files of a finer grain, each one storing all the actions that a certain learner of a specific group has performed during the shortest time interval (e.g. 1 minute).

Consequently, several of these log files can be concatenated so as to obtain the appropriate degree of granularity and thus fit their size to the characteristics of the computational resources of a distributed environment such as Grid. The aim is to distribute the workload among the distributed nodes correctly and as a result to parallelize the data processing efficiently. Finally, the processed results are stored in a database in a way that they can be easily analyzed by statistical packages so as to extract statistics about the learner's absolute and relative number of contributions. At the end of the process, this obtained knowledge is presented to all the learners of the same group as structured feedback in the form of pie chart, histogram, etc.

As a result, by providing a quantitative and qualitative description of the others' contributions may influence group members' motivation and production and as a consequence improve their participative behaviour in both qualitative and quantitative levels.

3 A Grid Implementation Using PlanetLab

In order to implement our experiment (see [2] and [14] for details of the generic design and the technologies used), we first developed a simple processing routine in

Java, called *EventExtractor,* which runs on a single machine and extracts event information from the BSCW event log files. This application converts the event information into well-formatted data to be stored in a persistent support. Specifically, the *EventExtractor* first receives the BSCW log files as an input, then it extracts specific data about users, objects, sessions, etc., and finally it stores the extracted data in a persistent support as a data structure. However, when executing sequentially, the *EventExtractor* needs a lot of time to process such amount of event information and hence it is not possible to constantly process a large size of data log files in real time.

In this point, we show how the MW paradigm is appropriate for processing log files of group activity in a Grid environment since we have different degrees of granularity available. Furthermore, there is no need for synchronization between the worker processors as tasks are completely independent from one another [2]. To this end, we have written a minimal Grid implementation prototype using the Globus Toolkit 3 (GT3) and have deployed it on the Planetlab platform (see [14] for a detailed description of these technologies). The aim is to demonstrate the viability of a Grid implementation of the MW paradigm for our problem domain.

In order to test our Grid prototype we turned Planetlab into a Grid by installing the GT3's Grid service container in every sliver of our slice. Moreover, we implemented the worker as a simple Grid service and deployed it on the GT3's container of every sliver of our slice. On the other hand, we wrote a simple Java client playing the role of the master which by using a simple list scheduling strategy dispatches tasks to the workers by calling the operations exposed by the worker Grid services, as follows:

The **worker** Grid service (see Fig. 1) publishes an interface with only one operation, *processEvents*. The master calls this operation in order to dispatch a task to the worker. The worker can only do one of these operations at a time (no multithreading).

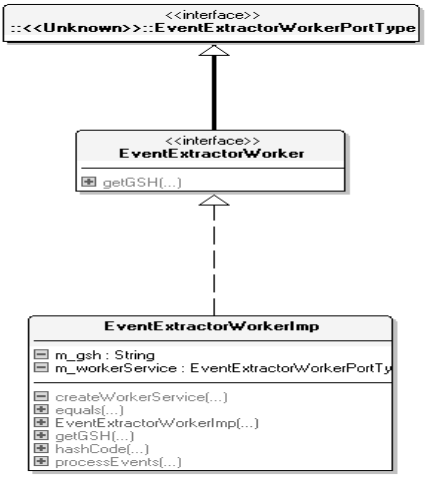


Fig. 1. The worker grid service implements the *EventExtractorWorker* interface which has only a single operation: `processEvents(String events, double dbInsertTimeInMs);` This operation parses the events passed so as to extract the required information and returns a data structure with performance information about the task executed

The operation has only one argument: a string containing the textual representation of the events to be processed by that task. The operation returns a data structure containing performance information about the task executed (elapsed time in ms, number of events processed and number of bytes processed). The *processEvents* operation is implemented by wrapping the Java code of the *EventExtractor* application's routine that parses the BSCW log events. In other words, the workers execute exactly the same java bytecode as the *EventExtractor* routine to process log events. This makes the performance comparison between the sequential and Grid approaches very sound.

On the other hand, the **master** (Fig. 2) is just a typical Java application that reads from a configuration file (1) the folder that contains the event log files to process, (2) the available workers, (3) the number of workers to use and (4) the size of the task to be dispatched to each worker expressed in number of events. The master then proceeds as follows: (1) peeks as much workers as needed from the configuration file and puts them all in a queue of idle workers, (2) enters a loop reading line by line (i.e. sensor component) the data contained in the event log files located in the folder specified in the configuration file, and (3) parses each one of these lines searching for the boundaries between events in order to extract those events (i.e. extractor component). Every time the master reads a number of events equal to the size of the task specified, it creates a thread that gets a worker from the queue of idle workers (synchronously waiting for a worker if the queue is empty) and synchronously calls the worker's *processEvent* operation. Once the call to the worker returns, the worker is put back into the queue of idle workers. The master exits the loop when all events in the event log files have been read and all the tasks that were dispatched (Fig. 3) have finalized.

Notice that the scheduling strategy (i.e. list scheduling) favors the speedier nodes and thus is appropriate for an environment where worker machines have unpredictable workloads as the Grid. However, in a more homogeneous workload environment a simple static round robin scheduling strategy could be more efficient.

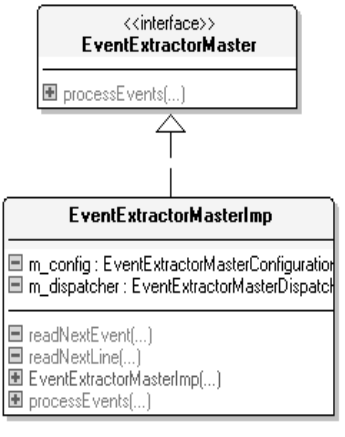


Fig. 2. The Master implements the *EventExtractorMaster* interface with a single operation to call the worker's *processEvents* operation and returns performance statistics about the execution. The *EventExtractorMasterImp* class aggregates an instance of *EventExtractorMasterDispatcher* to dispatch all tasks to available workers

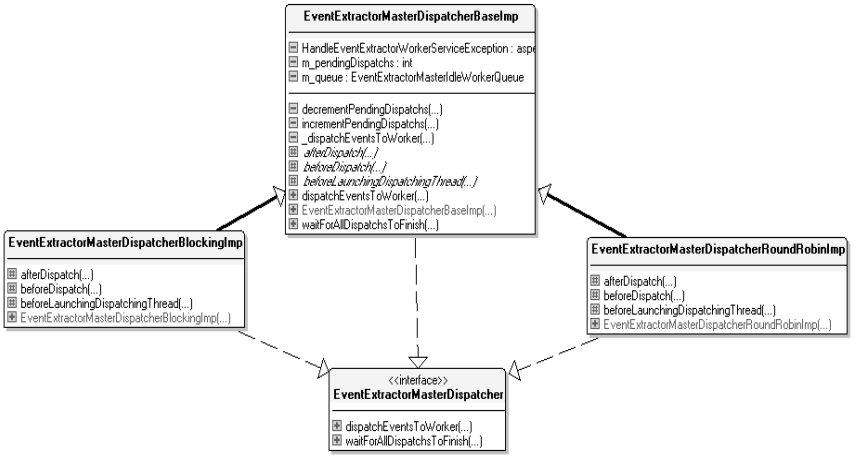


Fig. 3. Two strategies to dispatch tasks to workers: by blocking up to the queue of idle workers is empty, and by implementing the queue of idle workers with the *round-robin* scheme

4 Experimental Results

In order to carry out a comparative study between the sequential and Grid approaches, we designed a specific test battery in which we used both large amounts of event information and well-stratified short samples. Thus, on the one hand, in order to carry out certain tests we used all the existing daily log files making up the whole group activity generated during the spring term of 2004 in the course "Software Development Techniques" at the Open University of Catalonia. This course involved two classrooms, with a total of 140 students arranged in groups of 5 students and 2 tutors. On the other hand, other tests involved a few log files with selected file size and event complexity forming a sample of each representative stratum. This allowed us to obtain reliable statistical results using an input data size easy to use. All our test battery was processed by the *EventExtractor* routine executed in our Grid prototype on single-processor nodes involving usual configurations. The battery test was executed several times with different work load in order to have more reliable results in terms of statistical data involving file size, number of events processed and execution time along with other basic statistics.

The experimental results from the sequential processing are summarized in Figure 4 which presents the processing results of over one hundred event log files involving file size and processing time. This shows that the processing time is linear with respect to the number of events processed.

In this point, we show a sample of the main experimental results of our grid prototype (Fig. 5). Basically, they were obtained by running tests for different task sizes (i.e. in number of events) and number of workers (ranging in {2,4,8,16}) and observing efficiency and speed-up [14] for each set of workers.

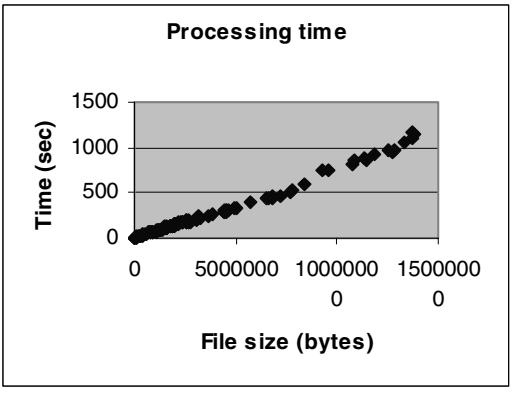


Fig. 4. Sequential processing time for each event log file size

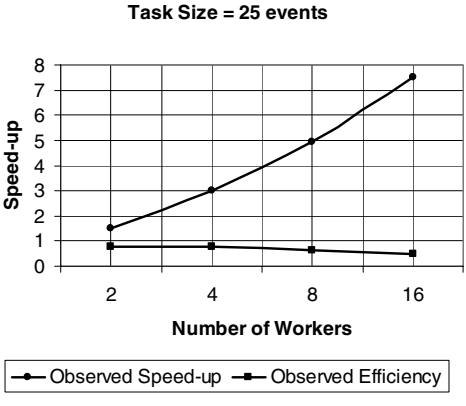


Fig. 5. Observed speed-up and efficiency for 25-event task and different number of workers

4.1 Analysis of the Results

From the results obtained it can be concluded that reasonable speed up has been achieved in every tested configuration. However, it is also true that the parallel efficiency decreases with the number of workers. We suspect this has to do with having fixed the size of the problem to 1000 events since the speed up seems to grow with the task size except for values near ceiling(1000/*n_workers*) where it begins to decrease. This is because for too small values of the task size the overhead introduced by the transmission protocol when sending the parts to the workers is noticeable and the implemented list scheduling strategy may be spending too much time waiting for completion notifications. In addition, values of the task size too near ceiling seriously diminish the attainable degree of concurrency, though here is where increasing the size of the task could be useful.

On the other hand, results of the parallelization were a little biased due to the homogeneous behaviour observed in Planetlab and they should be adjusted to the dynamic workload of a real Grid.

Finally, given that our test battery was based on the BSCW's log files, the results obtained were highly dependent on the low event complexity observed in BSCW's log data. We believe that event complexity is the cornerstone to take advantage of the powerful features provided by Grid.

5 Conclusions and Ongoing Work

In this paper, we have first argued how the provision of continuous feedback to online teams in CSCL environments can greatly improve the group activity in terms of interaction, problem-solving abilities, motivation and so on. To this end, large amounts of log information generated from the collaborative interaction need to be efficiently processed and analyzed. Moreover, in order to make the knowledge extracted from the analysis be useful for feedback purposes, users should be provided with both single information delivered fast in (almost) real-time and complex, exhaustive, yet structured deferred information, thus, stressing even more the processing requirements beyond those of a single computer.

Then, we proposed a Grid protoype to overcome these demanding requirements by improving the processing time of a large amount of complex event information of group activity log files and showed how this approach could be useful even for the case of BSCW log files which are limited in terms of complexity of information.

According to the results obtained in our study we conclude that it is viable to parallelize the structuring of group activity log data, achieving considerable speed up in the process. The question whether the Grid is beneficial or not will heavily depend on both the appropriate trade-off between task sizes and number of workers and the level of complexity of event log files to be processed. Therefore, these results encourage us to keep up working on the development of a real working Grid implementation to address the problem of processing efficiently group activity data from large log files.

As ongoing work, we plan to improve the communication between master and workers in our Grid prototype by exploring the convenience of using other features provided by the GT3 such as the Reliable File Transfer service and OGSI notification service. We believe that all these enhancements can not but increase the performance of our grid approach and that our current prototype, thanks to its simplicity, succeeds at establishing a lower bound on the expectation on the performance gain that we expect to achieve by grid-enabling our *EventExtractor* application. Moreover, we also plan to address other necessary improvements on our Grid prototype in future iterations such as fault-tolerance and dynamic discovery of available workers.

Acknowledgements. This work has been partially supported by the Spanish MCYT project TIC2002-04258-C03-03.

References

1. Dillenbourg, P. (ed.) (1999): Collaborative Learning. Cognitive and Computational Approaches. Elsevier Science Ltd. 1-19.
2. Xhafa, F., Caballé, S., Daradoumis, Th. and Zhou, N. (2004). A Grid-Based Approach for Processing Group Activity Log Files. In: proc. of the GADA'04, Agia Napa, Cyprus.
3. Zumbach, J., Schönemann, J., & Reimann, P. (2005). Analyzing and Supporting Collaboration in Cooperative Computer-Mediated Communication. In T. Koschmann, D. Suthers, & T. W. Chan (Eds.), Computer Supported Collaborative Learning 2005: The Next 10 Years! (pp. 758-767). Mahwah, NJ: Lawrence Erlbaum.
4. Zumbach, J., Hillers, A., Reimann, P. (2003). Supporting Distributed Problem-Based Learning: The Use of Feedback in Online Learning. In T. Roberts (Ed.), Online Collaborative Learning: Theory and Practice pp. 86-103. Hershey, PA: Idea.
5. Barros, M., Verdejo, M. (2000). Analysing student interaction processes in order to improve collaboration. The DEGREE approach. International Journal of Artificial Intelligence in Education, 11, 221-241.
6. T. Daradoumis, A. Martinez and F. Xhafa (2004). An Integrated Approach for Analysing and Assessing the Performance of Virtual Learning Groups, 10th Int. Workshop on Groupware, CRIWG'04, San Carlos, Costa Rica. Lecture Notes in Computer Science, Vol. 3198, pp. 289-304.
7. Caballé, S., Daradoumis, T., Paniagua, C. and Xhafa, F. (2005) A Grid Approach to Provide Effective Awareness to On-line Collaborative Learning Teams. In: Proc. of the 1st International Kaleidoscope Learning GRID Special Interest Group Workshop on Distributed e-Learning Environments (DLE'05). Napoli (Italy).
8. Open University of Catalonia http://www.uoc.edu (web page as of July 2005).
9. Bentley, R., Appelt, W., Busbach. U., Hinrichs, E., Kerr, D., Sikkel, S., Trevor, J. and Woetzel, G. (1997) Basic Support for Cooperative Work on the World Wide Web. Int. J. of Human-Computer Studies 46(6) 827-846.
10. Avouris, N., Komis, V., Fiotakis, F., Margaritis, M., Tselios, N. (2003) On tools for analysis of collaborative problem solving. Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies (ICALT'03).
11. Hardings, J. (2003) An XML-based Query Mechanism to Augment Awareness in Computer-integrated classrooms. 11th International Conference on Artificial Intelligence in Education, Australia.
12. Foster, I. and Kesselman, C. (Eds) (1999) The Grid 2e, 2nd Edition Blueprint for a New Computing Infrastructure. Morgan-Kaufman
13. Caballé S., Xhafa, F., Daradoumis, T. and Marquès, J.M. (2004). Towards a Generic Platform for Developing CSCL Applications Using Grid Infrastructure. In: Proc. of the CLAG/CCGRID'04, Chicago, USA.
14. Paniagua, C., Xhafa, F., Caballé, S. and Daradoumis, T. (2005) A Grid Prototype Implementation for Real Time Processing of Group Activity Log Data in Collaborative Applications. In: Proc. of the 2005 PDPTA'05. Las Vegas. USA.

Caching OGSI Grid Service Data to Allow Disconnected State Retrieval

Alastair Hampshire and Chris Greenhalgh

School of Computer Science and IT, University of Nottingham,
Jubilee Campus, Wollaton Road, Nottingham, NG8 1BB, UK
{axh, cmg}@cs.nott.ac.uk

Abstract. The Open Grid Services Infrastructure (OGSI) defines a standard set of facilities which allow the creation of wide area distributed computing applications, i.e. applications which cross organisational and administrative boundaries. An earlier project demonstrated significant potential for OGSI to support mobile or remote sensors, which require the integration of devices which are wirelessly and therefore only intermittently connected to the fixed network. Further, there is significant potential for mobile clients, e.g. PDAs, to be used for data analysis. However, OGSI currently assumes the availability of a permanent network connection between client and service. This paper proposes the use of caching to provide improved access to the state of intermittently connected OGSI grid services. The paper also describes a prototype which has been implemented and tested to prove the viability of the approach.

1 Introduction

Grid Computing symbolises a vision in which users share and access compute resources, such as databases, processor servers and mobile devices, which are distributed across large geographical areas and span multiple organisational boundaries [Foster and Kesselman, 1999]. One approach to grid computing utilises a *service oriented architecture*, the Open Grid Services Architecture (OGSA), in which grid resources are represented by entities known as *grid services*. The Open Grid Services Infrastructure (OGSI) [Tuecke et. al., 2003] enables wide area distributed computing by defining an open standard set of facilities for interacting with these grid services.

An earlier project, in which the authors were involved, explored the extent to which OGSI could be used to support interaction with mobile sensing devices [Barratt et. al, 2003] for data collection and computation purposes. In particular, this work focused on supporting a wearable medical sensing device for everyday health monitoring purposes. Integrating this and other sensing devices with the grid allows them to make use of grid services to, for example, archive data and perform computational analysis. Further, mobile client applications could be used to access grid resource to perform data analysis; for example, the Reality Grid project uses an OGSI client on a PDA to perform computational steering of simulations in condensed matter physics, plasma physics, and fluid dynamics [Brooke, Eickermann and Woessner, 2003]. However, mobile devices usually make use of wireless networks which, due to incomplete network coverage, are typically only intermittently available. Therefore any grid clients or services running on a mobile device are only intermittently available.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 284 – 293, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Building on earlier work [Hampshire, 2004; Hampshire and Greenhalgh, 2005] which described a framework to extend OGSI to intermittently available network environments, this paper proposes the use of caching as a technique to allow access to the state of temporarily unavailable OGSI grid services and/ or from temporarily unavailable OGSI grid clients. A partial prototype implementation has been produced to demonstrate the viability of the approach. The work in this paper allows OGSI to better support both mobile and remote data collection and data analysis.

In the rest of this paper, section 2 firstly describes OGSI, next describes an earlier exploration into the potential for OGSI to support mobile sensing devices and finally, examines some of the current uses of caching. Section 3 describes our approach to caching service state, addressing issues such as how to populate the cache and how to determine the cache consistency. Section 4 describes the initial validation.

2 Related Work

2.1 The Open Grid Services Infrastructure

The Web Services Architecture (WSA) [Booth et. al., 2004] provides an open standard definition of a web service, a language neutral software component with a well defined interface which uses XML as a common data format, permitting Internet based information exchange. The Web Services Description Language (WSDL) is typically used to define the interface to a web service. OGSI extends the WSA with functionality such as state management, notifications and lifetime management. The Globus Toolkit 3 (GT3) [Sotomayor, 2004] is an implementation the OGSI standard.

The OGSI idiom uses grid factory services to create grid service instances. A client (or clients) then interacts with the grid service instance, which is able to maintain state data for the duration of the client(s) interaction(s). OGSI defines the use of a *Grid Service Handle* (GSH) and a *Grid Service Reference* (GSR). A GSH is a persistent handle to the service, but does not contain protocol or location information. The GSR is a transient network pointer with an associated lifetime, which can be used to locate and invoke the grid service. The GSH can be resolved to a GSR using a *Handle Resolver Service*.

OGSI *Service Data* (SD) is a collection of XML elements that represent the state of a grid service instance. Typically, the SD associated with a particular grid service will be specified in the service's WSDL description. One or more *Service Data Elements* (SDE) are associated with each item of SD for a particular grid service. The OGSI specification defines a set of operations as a common way for accessing, querying and updating service data:

- findServiceData: retrieve service data based on a query statement
  - queryByServiceDataNames: retrieve the service data element with a given name.
- setServiceData: set the value of service data
  - setByServiceDataNames: set the service data element with a given name
  - deleteByServiceDataNames: delete the service data element with a given name.

The variability of each item of SD is defined by its mutability attribute as follows:

- Static: The SD value is defined in the WSDL port type and cannot change.
- Constant: The SD value is defined on creation of the grid service and cannot change throughout the lifetime of the grid service.
- Extendable: A SDE, once added, cannot be removed or modified. Only new SDEs can be added.
- Mutable: SDE values can be added and removed at any time.

OGSI allows clients to register an interest in a particular piece of service data. All interested clients are then notified when a change to the service data value occurs.

2.2 Integrating Mobile Sensing Devices with the Grid

The Equator eScience program, in collaboration with the MIAS IRC explored the extent to which the grid could be used to support mobile sensing applications. Part of this process included the development of an OGSI-compliant toolkit, the MIAS-Equator toolkit, which provides generic mechanisms for exposing mobile sensing devices as OGSI grid services; the toolkit was subsequently used to support a wearable medical monitoring device [Barratt et. al, 2003] and an environmental monitoring device deployed on a fresh water lake in the Antarctic [Benford et. al., 2003]. The toolkit uses a grid service instance to represent individual sensors with SD to temporarily stores sensor measurements. Integrating mobile sensing devices with the grid allows them to make use services available on the fixed network for data storage, computation, etc. Wireless networks, such as WiFi are used to provide a communication channel to and from the mobile device. However, the intermittently connected nature of OGSI clients and services deployed on mobile devices was identified as a limitation in the use of OGSI to support mobile devices. The work described in this paper aims to alleviate the problem by exploiting caching as a technique to improve access to the state data in intermittently available network environments.

2.3 Caching

Caching was originally conceived as a mechanism to provide faster access to data which is time consuming or expensive to fetch; this is achieved by collecting a duplicate of the data at a location which takes less time to access than the original data store. For example, web-caching [Wessels, 2001] works by storing copies of html files at a location close to the client upon their initial request by a client application; subsequent requests for the cached html file can be returned by the web cache. This not only reduces the retrieval time for a web page, but also reduces the load on web servers. One of the main difficulties involved with using a cache is in determining if the cached item is consistent with the original version. Considering the web caching example above, what happens if the original web page is changed? A number of approaches to overcome this problem have been devised, such as caching items alongside a lease time after which the cached item is discarded.

Caching has also been used to support continued operation of client applications when in an off-line or disconnected mode; an example of such a system is the Coda distributed file system [Braam, 1998] which allows clients to continue accessing and updating files whilst off-line from the actual file system.

A number of research projects have explored the extent to which caching can be used to support mobility in service oriented architectures (SOA), predominantly focusing on the Web Services Architecture; at the time of writing no work specifically addresses SD caching in OGSI, especially for intermittently connected services. Almost all research into caching in SOAs addresses client side mobility (i.e. access to services on the fixed network from an intermittently connected device).

Terry and Ramasubramanian [Terry and Ramasubramanian, 2003] make use of a client side cache of request response pairs to previous web service invocations to support partial continued operation of off-line client applications. The approach aims to be transparent to both clients and services by caching responses from available web services at an http proxy on the client machine. When operating in disconnected mode, the cache will satisfy any requests for which a cached response is available. Any update requests or requests without a corresponding cached response are queued at the proxy, to be replayed when the web service becomes re-available. The approach relies on the web service cache being pre-populated by prior web service requests. Terry et. al. conducted an experiment using caching with the Microsoft .NET MyServices, a set of web services which store and allow access to personal information, e.g. contacts, personal profile, calendar, etc. The experiments demonstrated the successful application of caching to support disconnected operation of a web service client application.

Kureshy [Kureshy, 2004] asserts that businesses are becoming increasingly interested in mobile applications. Kureshy states that autonomous mobile client applications can be achieved by replicating a subset of the application's business data and business logic on the client device in the form of a *Service Agent*. Kureshy suggests that careful selection of which data and logic to replicate is essential to successful disconnected operation. Several approaches to replaying transactions initiated whilst off line are suggested: modify replicated data in place and synchronise on reconnection; queue requests and replay when service is available. Queued requests must be delivered reliably, e.g. using a queuing product with built in reliability or using some reliable RPC style interface. This approach suggests an optimistic concurrency model where changes to data are submitted with both the old and the new value, allowing the service to spot and reject duplicate updates.

In general, the approaches described in this section which make use of caching to support intermittent network availability in service oriented architectures focusing on intermittently connected clients, but do not support intermittently connected services. Further, none of the approaches specifically address OGSI characteristics such as SD mutability and generic SD access mechanisms (see section 2.1 for details).

3 Approach

OGSI SD is particularly well suited to caching because state retrieval requests are made explicitly using the findServiceData operation. This allows a cache to easily identify which invocation are state retrieval requests (as opposed to updates) and therefore which invocations can be returned by the cache. Similarly for responses, the cache can identify state retrieval responses which can potentially be cached.

SD with a mutability value of *static* or *constant* (see section 2.1 for further details) is particularly well suited to caching; since these items do not change, there is no concern that the data will become invalid and can therefore be cached once and retained for the lifetime of the grid service. Further, static SD for a given port type could be cached just once for every grid service that implements that port type. Extendable SD could be cached without concern as to whether the cached data has been changed on or removed from the service; however, the SD will still be inconsistent if new values have been added to the SD on the service. Caching mutable SD is more problematic because the SD could be removed or change after being cached. Therefore some policy for checking or predicting cache consistency is required.

There are many possible strategies for caching SD and clearly no one size fits all. The choice of caching strategy depends on the usage scenario. This paper proposes several alternative approaches to SD caching; it is left as a decision for the OGSI grid application implementers, administrators or users to decide which is the most appropriate strategy for their usage scenario. Specifically, this paper addresses a number of questions concerned with caching OGSI SD: which SD should be cached? (section 3.1); how to populate the SD cache? (section 3.2); how to check the consistency of the cache? (section 3.3); and where to cache the SD? (section 3.4).

3.1 Selecting Which SD to Cache

Caching all SD for a particular service will, in most scenarios, not only be excessive, but also unnecessary. Much of the SD may never be accessed and the act of caching the data will waste storage space and network bandwidth. Four strategies for selecting which service data to cache are proposed:

- **Client:** The client is responsible for selecting which SD items to cache, achieved by making an explicit request to the service. This is a good approach as the client is clearly well positioned to know which SD items are of most interest to itself. Further, a client could inform the cache when it is no longer interested in the SD. The drawback to this approach is that the requesting client may be the only client to which the specified SD is of interest; therefore, caching the SD may be unnecessary and waste storage space.
- **Service:** The grid service itself specifies which SD items should be cached. The service developer and administrators are likely to know how the service will be used and therefore which items are best cached. The drawback to this approach is that data may be cached when there are no active clients interested in the service and its associated data; however, this is unlikely as in OGSI, grid service instances typically exist only for the duration of a client(s) interaction.
- **Notification:** A standard OGSI grid client will not normally be written to explicitly declare its interest in SD items. Instead, a service could interpret a SD notification subscription as a signal that the specified SD is of special interest and should therefore be cached. Because no explicit cache request call need be made, existing clients can make use of caching without requiring modifications.
- **Service Data Request:** A normal find service data request could be interpreted as a request to cache that SD item. This approach supposes that SD that is accessed once is likely to be accessed more often.

3.2 Caching Service Data Elements

Two possible approaches to populating an SD cache are described below:

- **Speculative:** a cache retains responses to successful find service data requests based on the assumption that if one client has requested a particular SDE it is likely that another client will request the same SDE.
- **Proactive:** the cache and service are responsible for maintaining an up-to-date copy of any SD which should be cached.

3.2.1 Speculative Caching

The SD cache stores a list of grid services and a corresponding list of SD items to cache. The SD cache is populated by collecting responses to successful find service data request. Speculative caching therefore makes the assumption that because a piece of SD has already been requested, it is likely to be requested again.

When an invocation is received, the cache handler is responsible for checking 3 things: that the target corresponds to a grid service in the cache list; that the invocation is a find service data request; and that the request is for a SD item which should be cached. If all three hold true, then the handler will check the cache for a response corresponding to the requested service and SD item. If no cached response is available then the service must be invoked in the normal fashion. If a cached response is available, then the cache handler may choose to immediately return the cached response or may first attempt to invoke the service and only return the cached response if the service invocation fails.

3.2.2 Proactive Caching

The SD cache and service are responsible for maintaining an up-to-date copy of the specified SD. This could be achieved by registering to receive notifications from the grid service when the SD of interest changes. When a find service data request is made for a cached SD item of an intermittently connected service, a cached copy of the SD can be returned immediately. No call to the grid service is necessary since the cache should contain the most up-to-date SDE which is available. However, if the service has been unavailable for some time, the cached SDE may have become out of date. Therefore, a strategy must be employed to decide if the cached response is still valid or if the call should be stalled to await service reconnection.

Because this approach requires additional communication to populate the cache (as apposed to speculative caching which requires no additional communication) additional network bandwidth is required. If the cache access frequency is low and/ or the SD changes frequently, considerable additional bandwidth could be used.

3.3 Cache Validity

The main problem with caching is in managing the consistency between the cached SDE and the actual SDE available on the service. This could be achieved by observing all invocations of the grid service for which the SD cache has been collected; given knowledge about which grid service operations change which SD items, the cache could be marked as inconsistent when invocations which change the cached SD are made. The problem with this approach is that it assumes that all invocations of a

particular grid service are made via the cache. The grid service could be invoked 'normally', i.e. without going via the cache. Further, the modification of SD may be internal to the service, for example a sensor service taking a new measurement.

The approach to cache consistency proposed by this paper involves annotating the cached SD with meta-data to aid the decision about cache validity. As already stated, data marked with a mutability attribute of *static* or *constant* cannot change on the server and therefore can safely be returned to the client. Data marked as *extendable* or *mutable* could be modified on the server and therefore a strategy for determining the validity of the cached data is required. The age of the cached SD could be calculated by comparing a timestamp denoting when the cache was made with current time. Cached SD items under a given age could be considered valid. Because, some SD will change more often than others, knowledge about how often a SD item is likely to change is required. This data could be provided in one of the following ways:

- The service administrators are responsible for specifying how often the SD is likely to change and therefore how long a cached SDE will remain valid. A conservative estimate is recommendable. Further, the cached SDE should be considered valid only for some fraction of the estimated value.
- The service collects information about how often a SDE changes. Over the lifetime of the service, detailed information about access and update patterns would be collected. This information could then be used in the same way as above to decide if a SDE is valid. Advanced techniques could be used to spot, for example, that a service is typically accessed very infrequently during a certain time period and more frequently during a different time period.
- The SD may change at regular, predetermined intervals. For example, considering a grid service interface to a sensing device with SD representing measurements, the SD value will change every time the sensor takes a reading. The sensor may be preconfigured to take readings at predefined time intervals, e.g. every hour. Given knowledge about when the reading last changed and the current time, the cache can safely infer if the SD value has changed.

Some SD items may not need to be 100% up-to-date to be useful to the client. For example, considering the scenario of a grid service used to represent and interface to a mobile sensing device (see section 2.2 for details) the client may still be interested in sensor readings even if they are not the latest measurements. The client could collect any new readings when the service becomes re-available. This kind of meta-information could also be used to help make a decision on the validity of cached SD.

As already stated, when a cached SD is requested, a decision must be taken as to whether the cached item should be used, i.e. is sufficiently likely to be consistent with the actual service value. This decision can be made in one of two places:

- **By the client:** When returning a SD request from the cache, meta data could be added to the response to alert the client to the fact that this response is a cached response and not a direct response from the service. The meta-data could contain a timestamp indicating how long the item had been cached for and possibly some metrics to indicate for how long the SD in question typically remains valid (as discussed above). The client will then be responsible determining if the response is acceptable. The client may decide to re-request the SD, this time explicitly specifying that the response should not come from the cache.

- **By the cache:** The cache is responsible for making a decision about whether the cached response is still valid. If the cache believes the cached response is no longer valid then the request must be forwarded to the service in the normal manner. If the cached item is believed to be valid, the cache could still return meta-information with the response to notify the client that this is a cached response. The client may decide that the cached response is too old and remake the request anyway.

3.4 Where to Cache Service Data

For the purpose of supporting disconnected operation, the best place to cache SD is just before the intermittently available network hop. Caches placed closer to the client could be used to improve access time to SD, however this paper deals strictly with issues related to mobility and intermittent network connectivity, therefore a discussion of how caching could be used to provide faster access to OGSI SD is outside the scope of this document.

This paper proposes two different caching scenarios: caching SD to allow intermittently connected clients improved access to the SD of permanently connected OGSI services and caching SD to allow permanently connected clients improved access to the SD of intermittently connected OGSI services.

3.4.1 Intermittently Connected Service

To support continued access to the SD of an intermittently connected grid service, a SD cache should be made available somewhere on the fixed network. The best place for the SD cache would be at the access point to the intermittently connected network.

3.4.2 Intermittently Connected Client

Depending on the programming style used, a client may never re-request the same SD twice, so in this case, using speculative caching may not be very helpful. However, if developer use grid services like distributed objects, then clients may repeatedly re-request items of SD, making speculative caching more useful. Proactive caching of non-constant SD items will not scale to large numbers of clients as the grid service will have to send notifications to each client. Where a group of client applications exist within a small geographical area, peer-to-peer techniques could be used; find service data requests could be honoured by other nearby clients which have recently requested the required SD.

4 Validation

A partial implementation of the caching approaches described in section 3 has been developed; specifically, proactive caching has been implemented (see section 3.2 for more details of caching types). A SD cache deployed on a permanently available machine contains a hash-table of cached SD items, indexed by the service's GSH and the SD name. For each cached SD item, the SD value is stored alongside a timestamp recording when the cache was made. The Web Services Routing Protocol (WS-RP) is used to route invocations via the SD cache. The cache component listens for SD

notifications that correspond to the SD being cached, which it uses to populate the cache. When a find service data invocation is received, the cache first attempts to contact the service directly. Should this invocation fail, the cache will return a corresponding cached response if one is available. When returning a cached response, the cache inserts a soap header element "CachedResponse" containing a value, in milliseconds, indicating how old the cached item is.

The above described prototype implementation has been used to support the operation of the MIAS-Equator toolkit described in section 2.2. A simple test demonstrated the cache being successfully populated each time new sensor measurements were taken. Further, a client application was shown to be able to retrieve SD from a service whilst the service was temporarily unavailable.

5 Conclusion

Mobile and therefore intermittently connected sensing devices could be integrated with the traditional grid infrastructure to allow archival and processing of collected sensor data. Further, mobile or remote OGSI grid clients, e.g. running on a PDA, could be used to visualise, monitor or steer data analysis. However, standard OGSI does not currently support the deployment of clients or service on devices which are only intermittently connected to the fixed network.

This paper has described how SD caching could be used to support access to the SD of temporarily disconnected services and SD retrieval from temporarily disconnected OGSI client applications; this technique will allow OGSI to better support the applications described above. It has been shown that some SD, e.g. SD denoted as either *static* or *constant*, is particularly well suited to being cached. This paper has described: techniques for selecting which SD items to cache; procedures to populate a SD cache; techniques for ascertaining the validity of cached SD items; and guidance on where to place a SD cache. It is acknowledged that no one caching scheme fulfils all requirements and it is left as a choice for the application administrator or implementer to choose the most appropriate approach, guided by the considerations identified in this paper. A prototype has been implemented and shown to successfully allow access to the SD of temporarily unavailable OGSI grid services.

References

Barratt, Carl, et al. (2003), "Extending the Grid to Support Remote Medical Monitoring", UK e-Science All Hands Meeting 2003.

Booth, David, Haas, Hugo, McCabe, Francis, Newcomer, Eric, Champion, Michael, Ferris, Chris and Orchard, David (2004), "Web Services Architecture", W3C Working Group Note, February 11, 2004.

Braam, Peter J. (1998), "The Coda Distributed File System", Linux Journal #50, 1998.

Brooke, John, Eickermann, Thomas and Woessner, Uwe (2003), "Application Steering in a Collaborative Environment", Proc. 2003 ACM/IEEE conference on Supercomputing (SC'03), Phoenix, Arizona, USA, November 15-21, 2003.

Foster, Ian and Kesselman, Carl (1999), "The Grid: Blueprint for a New Computing Infrastructure", Morgan Kaufmann, 1999.

Hampshire, Alastair (2004), "Extending the Open Grid Services Architecture to Intermittently Available Wireless Networks", UK e-Science All Hands 2004.

Hampshire, Alastair, Greenhalgh, Chris (2005), "Supporting OGSI and WSRF Specific Functionality in Intermittently Available Network Environments", UK e-Science All Hands 2005.

Kureshy, Arif (2004), "Architecting Disconnected Mobile Applications Using a Service Oriented Architecture", MSDN Library, September, 2004.

Sotomayor, Borja (2004), "The Globus Toolkit 3 Programmer's Tutorial" http://gdp.globus.org/gt3-tutorial/singlehtml/progtutorial_0.4.3.html (verified 12/05/05).

Terry, Douglas B. and Ramasubramanian, Venugopalan (2003), "Caching XML Web Services for Mobility", ACM Queue, May, 2003, pp. 71-78.

Tuecke, S., Czajkowski K., Foster, I., Frey, J., Graham, S., Kesselman, C., Maguire, T., Sandholm, T., Vanderbilt, P. and Snelling, D. (2003), "Open Grid Services Infrastructure (OGSI) Version 1.0". Global Grid Forum Draft Recommendation, 2003.

Wessels, Duane (2001), "Web Caching", O'Reilly & Associates Incorporated, ISBN: 156592536X, July, 2001.

Shelter from the Storm: Building a Safe Archive in a Hostile World

Jon MacLaren, Gabrielle Allen, Chirag Dekate, Dayong Huang, Andrei Hutanu, and Chongjie Zhang

Centre for Computation and Technology, Louisiana State University, Baton Rouge, LA 70803
{maclaren, gallen, cdekate, dayong, ahutanu, czhang}@cct.lsu.edu

Abstract. The storing of data and configuration files related to scientific experiments is vital if those experiments are to remain reproducible, or if the data is to be shared easily. The prescence of historical (observed) data is also important in order to assist in model evaluation and development. This paper describes the design and implementation process for a data archive, which was required for a coastal modelling project.

The construction of the archive is described in detail, from its design through to deployment and testing. As we will show, the archive has been designed to tolerate failures in its communications with external services, and also to ensure that no information is lost if the archive itself fails, i.e. upon restarting, the archive will still be in exactly the same state.

1 Introduction

The Southeastern Coastal Ocean Observing and Prediction (SCOOP) Program's Coastal Model Project [15], is an ongoing collaboration between the modeling research community and operational agencies, such as the National Oceanic and Atmospheric Administration (NOAA). The project aims to take today's cutting-edge activities from the research community, and develop these so they can form the basis for tomorrow's operational systems.

Part of this project's work involves the regular execution of coastal modeling codes, such as ADCIRC [13] and SWAN [11], for various geographical regions. Additional runs of some codes are performed to predict the path and effects of ongoing tropical storms and hurricanes; the results from these runs are passed to groups involved in evacuation planning. The project also tries to verify the accuracy of the coastal models by verifying the predictions the codes make against real-world observed data.

To support this work, a data archive was required which would store:

- atmospheric model outputs (wind data),
- results generated by the hydrodynamic models, which use the atmospheric model outputs for input (wage/surge data), and
- observational data to be used for verification of model results (sensor data).

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 294–303, 2005.
© Springer-Verlag Berlin Heidelberg 2005

The archive would therefore form a backbone for the research efforts of the project, and as such, have to be both highly available, and reliable.

To meet this need, a data archive was constructed at the Center for Computation and Technology. Although supporting the SCOOP Coastal Model Project was our prime objective, we wanted to be able to re-use most of the archive's functionality, and code, for other efforts with data storage requirements, e.g. our group's numerical relativity work.

This paper describes the construction of this archive in detail. Section 2 briefly discusses data storage requirements, then describes the design of the archive service; Section 3 describes the archive's implementation. Section 4 describes how APIs and tools were developed to allow easy access to the archive, and Section 5 explains how good engineering practices were used to ensure reliability and code re-use. Finally, Section 6 explains some ideas for future work, and Section 7 gives our conclusions.

2 Design and Architecture

The data storage requirements for the SCOOP Project are simple. The files to be stored are small (no more than a few MB at most) so they can be easily moved to the archive.[1] In addition, there was no requirement to provide any kind of access control.

To complement the archive, a Metadata Catalog would be provided, which would store information about the model configurations used to generate the stored data. This catalog should be the first port of call for people looking for data outputs from the project, and can provide references to the location of the data, in response to users' searches. As the catalog is not the subject of this paper, it is not described here, although interactions between the archive and the catalog are.

The architecture for the archive is shown in Figure 1. In more detail, the steps for uploading files to the archive are as follows:

U1. The client contacts the archive, providing a list of files which they wish to upload to the archive. The archive decides where each file will be located within the archive.[2] The archive's response groups the original files into one or more **transactions**, each of which is associated with: a location in the storage area (specified by a set of URLs for various scheme names); a subset of the list of files; and an identifier which the client uses in later interactions.

[1] At the time we began construction of the archive, it was not clear to us what volume of data would be generated by the project each day, nor was it clear how long data needed to be kept for. We have since discovered that the project generates approximately 1 TB of data per month.

[2] Within the SCOOP Project, there is a File Naming Convention, allowing the archive to deduce metadata from the names of the files, and thus determine the location of each file within the archive's directory structure. Files belonging to the output of a single run of a model code will be stored in the same directory. New directories for new code runs are created automatically.

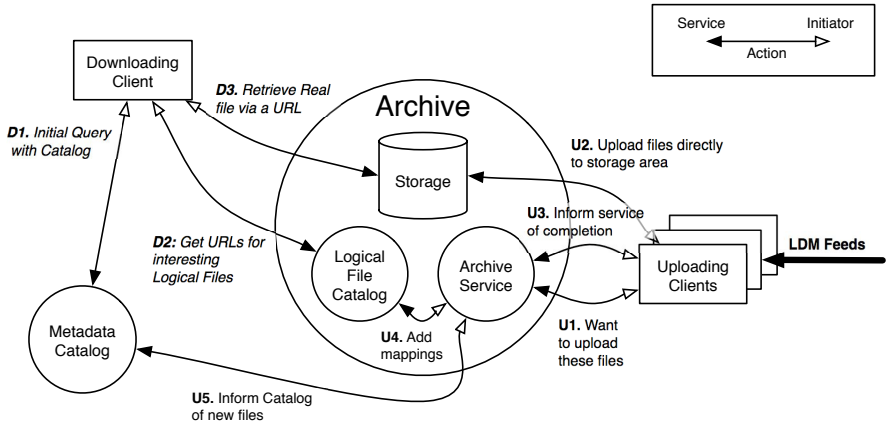


Fig. 1. Basic architecture for the SCOOP Archive, showing steps for **U**ploading and **D**ownloading files

The following steps are then carried out for each of the transactions.

U2. The client uploads the files to the archive storage with some third-party software, e.g. GridFTP [8,2], and a URL given to it by the archive.

U3: request. After the client has uploaded the files to the location, it informs the archive that the upload is complete (or aborted), using the identifier.

U4. The archive contructs Logical File Names (LFNs) for the files which have been uploaded, and adds mappings to Logical File Catalog that link the LFNs to various URLs that can be used to access the physical files.

U5. The archive informs the catalog that there are new files available in the archive, providing the LFNs.

U3: response. The archive returns the LFNs to the client.

The steps for downloading a file from the archive are as follows:

D1. A client queries the **Metadata Catalog** to discover interesting data, e.g. ADCIRC Model output for the Gulf of Mexico region, during June 2005. Some Logical File Names (LFNs) are returned as part of the output, together with a pointer to the Archive's Logical File Catalog.

D2. The client chooses some LFNs that they are interested in, and contacts the Logical File Catalog service to obtain the files' URLs.

D3. The client picks URLs (based on scheme name, e.g. `gsiftp` for GridFTP) and downloads the files directly from the Archive Storage.

Note that this simply describes the sequence of interactions that are exchanged in order to achieve these tasks. We will later show that the clients indicated in the diagram can be either command-line tools, or a portal web-page.

3 Implementing the Archive Service

The architecture was refined into a message sequence diagram, and implemented using Web Services. Early on, we chose to use the Grid Application Toolkit (GAT) [4], developed as part of the GridLab project [3], to help with file movement, and also to provide access to Logical File services. The GAT wraps different underlying technologies (using "adaptors"), and so provides consistent interfaces. Here, the underlying Logical File Catalog is a Globus RLS, but this could be replaced with a similar component, without changing the archive's code. Our desire to use the GAT led us to select C++ as the language to use for implementing the service, which in turn led to us using the Web Service tooling provided by gSOAP [6,7]. Following the advice from the WS-I Basic Profile 1.1 [5, Sec. 4.7.4], we avoided using RPC/Encoded style[3] for our services. Instead we chose Document/Literal style, first designing the XML messages that would be exchanged, then producing XML Schema and WSDL definitions. From these definitions, the gSOAP tooling was used to automatically generate code stubs.

During the upload process, the archive passes back URLs for a staging area, rather than allowing clients to write directly to the Archive Storage. This also makes it simpler to prevent additional (i.e. unauthorized) files from being inserted into the archive. A distinct staging directory is created for each transaction identified by the archive.

4 Archive Interfaces and Tools

4.1 Downloading

We have provided two complementary mechanisms for clients to download data, namely:

- Command-line tools, e.g. `getdata`; and
- A portal interface, built using GridSphere [9], an open-source portal framework,[4], also an output from the GridLab project [3], which uses JSR 168 compliant portlets [1].

The `getdata` tool has a simple syntax, encapsulating the client side of the message exchanges with the Logical File Service and the download from the Archive Storage, and can choose between different protocols for downloading the data. This was achieved using the GAT, making `getdata` easily extensible if new protocols need to be added. Currently, GridFTP and https downloads are supported.

Through the portal interface, users can access the same functionality as with the command-line tools. Users can search for files, and download them, either

[3] **Restriction R2706** (which also appeared in Basic Profile 1.0) states: "A `wsdl:binding` in a DESCRIPTION MUST use the value of "literal" for the `use` attribute in all `soapbind:body`, `soapbind:fault`, `soapbind:header` and `soapbind:headerfault` elements."

[4] Available for download from `http://www.gridsphere.org/` at time of writing.

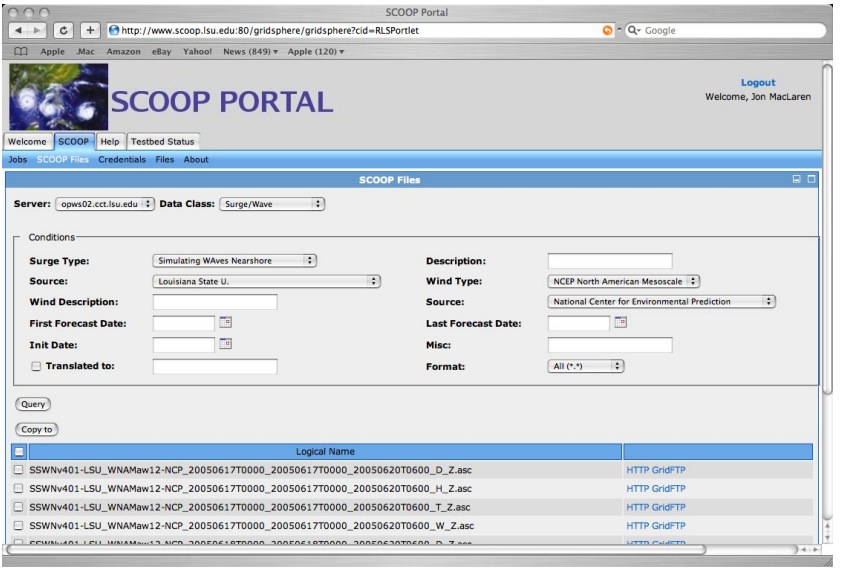


Fig. 2. Screen capture from the SCOOP Portal, showing a typical search

through the browser, or perform a third-party file transfer via GridFTP. The portal interface, shown in Figure 2, integrates this and other capabilities, such as Grid monitoring, Job Submission and Visualization into a single interface.

4.2 Uploading

In order to simplify the construction of clients, a two-layered C++ client API was written and, like the service, was based on gSOAP. The first level of the API neatly encapsulates each of the two message exchanges, labeled **U1** and **U3** in Figure 1, into two calls `start_upload` and `end_upload`.

A higher-level API with a single call, `upload`, is also provided. This encapsulates the entire process, including the uploading of the files themselves. The `upload` call has the following signature:

```
bool upload(std::vector<std::string>& uploadFiles,
            std::string               urlType,
            bool                      verbose);
```

The provision of such a layered API makes the construction of tools far simpler.

Currently, files can only be uploaded via a command-line tool, `upload`, which allows a variety of transport mechanisms to be used to copy files into the archive. Even though the interactions with the archive service are more complicated than for downloading files, the command-line syntax remains trivial.

```
upload -gsiftp -done rm SSWN*.asc
```

This example will upload all ".asc" files in the current directory starting "SSWN" (produced using the SWAN code), transferring the files with GridFTP (GridFTP

uses URLs with the scheme name "gsiftp") and will remove the files if they are successfully uploaded (specified by "-done rm").

5 Ensuring Stable, Robust, Re-usable Code

A key challenge when building distributed systems is tolerating problems with connectivity to external services. However, it would also be reckless to assume that our archive service would be perfectly reliable. No feasible amount of testing is sufficient to remove all the bugs from even a moderately sized program. In addition to the archive code, we are also relying upon the GAT and gSoap, (not to mention the Linux C library, compilers and operating system).

We have employed a number of techniques while designing the archive to make it as reliable as possible. These techniques, plus other "good practices" that we have employed, are described below.

5.1 Tolerating Failure in Remote Services

In the architecture shown in Figure 1, the Metadata Catalog is clearly a distinct component. However, in our implementation, from the perspective of the Archive Service component of the Archive, both the Metadata Catalog and the Logical File Catalog are remote components; only the Storage component is physically co-located with the Archive Service.

As stated earlier, remote components may become unavailable temporarily, only to return later. These **partial failures** are not encountered in "local" computing. If you lose contact with a program running on your own system, it is clear that some sort of catastrophic failure has occured; this is not the case in a distributed system. For an excellent discussion on partial failures and other inherent differences between local and distributed computing, the reader is directed to [12].

Here, we have tried to insulate ourselves from partial failure as far as possible. Following the advice in [12], we have made the partial failure explicit in the archive service interface. In the response part of interaction **U3**, when the user is returned the set of logical files corresponding to the newly uploaded files, they are told which of these logical files have been successfully stored in the Logical File Catalog.[5] The user knows that they need take no further action, and that the logical files will be uploaded to the Logical File Catalog when it becomes available again.

5.2 Recovering from Failures in the Archive

Interactions with the Archive are not stateless. Transaction IDs are created, and associated with areas in the Archive Storage, and with files that are to be uploaded. These IDs are used in later interactions and must be remembered by the Archive if it is to function correctly.

[5] Similarly, they are informed of whether or not the files have been registered with the Metadata Catalog.

Given what was stated earlier about the reliability of our own programming, and our operating environment, we chose to place all such state into a database located on the machine with the Archive Service. The "pending" insertions for the Logical File Catalog and Metadata Catalog (described in the previous section) are also stored in this database. Thus, if the service terminates for some reason, and restarted, it is able to pick up from exactly where it left off.

Note that we can also correctly deal with partial failure in the case where a transaction might be completed, but the response fail to reach the client. The client can safely retry the complete/abort transaction operation until they receive a response. If they receive a message stating that the complete/abort has succeeded, then they have just now terminated the transaction. If they receive a message stating that the transaction is unknown, then a previous attempt to complete/abort the transaction must have succeeded.

5.3 Keeping Domain-Specific Code Separate

Although the archive was primarily created for use in the SCOOP project, we have tried to keep project-specific functions separate from generic functions. Specifically, SCOOP uses a strict file naming convention, from which some of the file's metadata may be extracted. The filename therefore dictates where the file should be stored, etc. To keep the project-specific code separate, methods on a `FilingLogic` object are used to decide where to place all incoming files. Different subclasses of the `FilingLogic` class can be implemented for different "flavours" of archives.[6]

5.4 Summary

Through extensive testing, we have determined that the archive is stable. During initial trials, we used multiple clients to simultaneously upload files in rapid succession. Over one weekend, 20,000 files were successfully uploaded. The archive remained operational for a further three weeks (inserting a further 10,000 files), until a change to the code necessitated that it be manually shutdown.

During this time, we monitored the size of the Archive Service process. It seems that the program does leak a small amount of memory. After a number of months, this would likely cause the process to fall over. To prevent this happening, we have chosen to manually shut the service down every 14 days, and then restart. This "preventative maintenance" ensures that the archive does not fail unexpectedly.[7]

Although we have strived to make the archive as reliable as possible, there is a limit to how much we can improve the availability of the archive while it still resides on a single machine. The hardware in the archive machine is not perfect,

[6] Undoubtedly when the archive is first applied to a new project, there will be new requirements, and the FilingLogic interface will change. Nonetheless, this transition will be greatly simplified by the existence of this boundary.

[7] If for some reason, the archive needed to remain operational during the scheduled maintenance time, this could easily be moved or canceled (provided many successive shutdowns are not canceled).

nor are we using an Uninterruptable Power Supply (UPS). The campus network also causes periodic failures.

It seems that replicating the data archive would yield the biggest improvements in reliability.

6 Future Work

This first version of the archive provides us with a useful basis for future development. There are a number of ways in which we want to extend the basic functionality described above, the two most important of which are explained below.

6.1 Transforming Data on Upload/Download

Currently, the archive stores data in the form in which it is uploaded; downloading clients receive the data in this same format. We wish to support the following scenarios:

- The compression of large ASCII-based files when they enter the archive, and their decompression when they are downloaded (preferably after they have reached the client).
- The partial retrieval of a dataset. Some of the data stored in the archive is in NetCDF format [14], which supports retrieval of subsets of variables, ranges of timesteps, etc.
- Retrieval of data in different formats, e.g. retrieving single precision data from a double precision file.

To support this type of operation, we are proposing to associate a **specification** with each file that specifies the current format which the file is in, the type of compression, etc. Specifications are used at upload and download time; files may be transformed by the archive upon arrival.

6.2 Notification

One of the key goals of the SCOOP Project is to improve responsiveness to storm events, such as hurricanes, which are relatively common in the Southern United States. When a hurricane advisory arrives at the archive, it should trigger high-priority forecasts for the current location of the storm.

To support this work, we have recently implemented a simple interface that can be built upon to perform sophisticated patterns of notification. When a file is ingested into the archive, a message is sent to the `FilingLogic` object. The SCOOP implementation of this executes a script (forked to run in the background, so as to not affect the archive's performance), passing the Logical and Physical File Names as parameters.

6.3 Lifetime Management for Data

Currently, data is removed from the archive automatically after a fixed time. It should be possible for uploading clients to request storage for a particular duration. It should also be possible for this lifetime to be altered by other, authorized clients.

7 Conclusions

We have described the construction of a reliable data archive, constructed to satisfy storage requirements from a coastal modeling project. A number of techniques were employed, from the design phase through to the final testing, to ensure reliability.

We also showed how the archive was designed so that it could be re-used in other projects. In particular, we endeavoured to keep all project-specific code separate from the generic code, and provided an internal API which allows new project-specific code to be easily provided.

It is likely that future versions of the archive will rely on other systems for backend data storage. The most obvious candidate is the Storage Resource Broker (SRB) from SDSC [10], which provides excellent support for managing highly distributed data stores, and which would also satisfy some of our new requirements from Section 6, e.g. the retrieval of subsets of data.

Acknowledgments

This work was made possible by funding from the Office of Naval Research and the National Oceanic and Atmospheric Association, received through Louisiana State University's participation in the Southeastern Universities Research Association (SURA) Southeastern Coastal Ocean Observing and Prediction (SCOOP) program.

References

1. A. Abdelnur, S. Hepper, *Java^{TM} Portlet Specification Version 1.0*, Java Specification Request 168 (JSR 168), Commuinity Development of Java Technology Specifications, Oct. 2003. Online at: `http://jcp.org/en/jsr/detail?id=168`
2. W. Allcock (Ed.), *GridFTP: Protocol Extensions to FTP for the Grid*, Global Grid Forum Recommendation Document GFD.20. Online at: `http://www.ggf.org/documents`
3. G. Allen *et al*, "Enabling Applications on the Grid: A GridLab Overview", in *International Journal of High Performance Computing Applications*, Vol. 17, No. 4, SAGE Publications, Nov. 2003, pp. 449–466.
4. G. Allen *et al*, "The Grid Application Toolkit: Toward Generic and Easy Application Programming Interfaces for the Grid", *Proceedings of the IEEE*, Vol. 93, No. 3, 2005, pp. 534–550.
5. K. Ballinger, D. Ehnebuske, *et al* (Eds.), *Basic Profile Version 1.1 (Final)*, The Web Services-Interoperability Organization (WS-I),Aug. 2004. Online at: `http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html`
6. R. van Engelen, K. Gallivan, "The gSOAP Toolkit for Web Services and Peer-to-Peer Computing Networks", in *2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2002)*, IEEE Press, 2002, pp. 128–135.
7. R. van Engelen, G. Gupta, S. Pant, "Developing Web Services for C and C++", in *IEEE Internet Computing*, Vol. 7, No. 2, pp. 53–61, Mar./Apr. 2003.

8. I. Mandrichenko (Ed.), W. Allcock, T. Perelmutov, *GridFTP v2 Protocol Description*, Global Grid Forum Recommendation Document GFD.47. Online at: `http://www.ggf.org/documents`
9. J. Novotny, M. Russell, O. Wehrens, "GridSphere: a portal framework for building collaborations", in *Concurrency and Computation: Practice and Experience*, Vol. 16, No. 5, Wiley, Apr. 2004, pp. 503–513.
10. A. Rajasekar, M. Wan, R. Moore, "MySRB and SRB - Components of a Data Grid", in *Proceedings of the 11th International Symposium on High Performance Distributed Computing (HPDC-11)*, Edinburgh, Scotland, pp. 301–310, July 2002.
11. R. C. Ris, L. H. Holthuijsen, N. Booij, "A Spectral Model for Waves in the Near Shore Zone", in *Proceedings of the 24th International Conference on Coastal Engineering*, Kobe, Oct. 1994, Japan, pp. 68–78, 1994.
12. J. Waldo, G. Wyant, A. Wollrath, S. Kendall, *A Note on Distributed Computing*, Technical Report TR-94-29, Sun Microsystems Laboratories, Inc., Nov. 1994. Online at: `http://research.sun.com/techrep/1994/abstract-29.html`
13. J. Westerink, R. Luettich, A. Baptista, N. Scheffner, and P. Farrar, "Tide and Storm Surge Predictions Using Finite Element Model", in *ASCE Journal of Hydraulic Engineering*, pp. 1373–1390, 1992.
14. Unidata's Network Common Data Form (NetCDF). `http://my.unidata.ucar.edu/content/software/netcdf/index.html`
15. The SCOOP Program's Coastal Model Project Website. `http://www1.sura.org/3000/3310_Scoop.html`

Event Broker Grids with Filtering, Aggregation, and Correlation for Wireless Sensor Data

Eiko Yoneki

University of Cambridge Computer Laboratory,
Cambridge CB3 0FD, United Kingdom
eiko.yoneki@cl.cam.ac.uk

Abstract. A significant increase in real world event monitoring capability with wireless sensor networks brought a new challenge to ubiquitous computing. To manage high volume and faulty sensor data, it requires more sophisticated event filtering, aggregation and correlation over time and space in heterogeneous network environments. Event management will be a multi-step operation from event sources to final subscribers, combining information collected by wireless devices into higher-level information or knowledge. At the same time, the subscriber's interest has to be efficiently propagated to event sources. We describe an event broker grid approach based on service-oriented architecture to deal with this evolution, focusing on the coordination of event filtering, aggregation and correlation function residing in event broker grids. An experimental prototype in the simulation environment with Active BAT system is presented.

1 Introduction

Recent progress in ubiquitous computing with a dramatic increase of event monitoring capabilities by Wireless Sensor Networks (WSNs) is significant. Sensors can detect atomic pieces of information, and the data gathered from different devices produce information that has never been obtained before. Combining regionally sensed data from different locations may spawn further useful information. An important issue is to filter, correlate, and manage the sensed data at the right time and place when they flow over heterogeneous network environments. Thus, an integrated event correlation service over time and space is crucial in such environments.

Event correlation services are becoming important for constructing reactive distributed applications. It takes place as part of applications, event notification services or workflow coordinators. In event-based middleware systems such as event broker grids, an event correlation service allows consumers to subscribe to patterns of events. This provides an additional dimension of data management, improvement of scalability and performance in distributed systems. Particularly in wireless networks, it helps to simplify the application logic and to reduce its complexity by middleware services. It is not easy to provide reliable and useful data among the massive information from WSNs. Mining new information from sensed data is one issue, while propagating queries over WSNs is a different issue. Combination of both approaches will enhance data quality, including users'

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 304–313, 2005.
© Springer-Verlag Berlin Heidelberg 2005

intentions such as receiving, providing, and passing data. At the same time, data should be managed as openly as possible.

Middleware's Task: Middleware for sensor networks can be defined as software that provides data aggregation and management mechanisms, adapting to the target application's need, where data are collected from sensor networks. This functionality must be well integrated within the scheme of ubiquitous computing. The middleware should offer an open platform for users to seamlessly utilize various resources in physically interacting environments, unlike the traditional closed network setting for specific applications. As a part of this approach, mobile devices will play an important role and will be used for collecting sensor data over ad hoc networks, conveying it to Internet backbone nodes. Mobile devices can be deployed in remote locations without a network infrastructure, but they are more resource constrained, and a detectable/implementable event detection mechanism is required.

The trend of system architecture to support such platforms is towards service broker grids based on service management. When designing middleware for sensor networks, heterogeneity of information over global distributed systems must be considered. The sensed information by the devices is aggregated and combined into higher-level information or knowledge and may be used as context. The publish/subscribe paradigm becomes powerful in such environments. For example, a publisher broker node can act as a gateway from a WSN, performing data aggregation and distributing filtered data to other networks based on contents. Event broker nodes that offer data aggregation services can efficiently coordinate data flow. Especially with the distributed event-based middleware over peer-to-peer (P2P) overlay network environments, the construction of event broker grids will extend the seamless messaging capability over scalable heterogeneous network environments. Event Correlation will be a multi-step operation from event sources to final subscribers, combining information collected by wireless devices into higher-level information or knowledge.

There has been much effort for in-network data aggregation such as TinyDB [8]. However, a mainstream of deployments of sensor networks is to collect all the data from the sensor networks and to store them in database. Data analysis is preceded from the data in the database. We propose a distributed middleware architecture integrating global systems to support high volume sensor data. We prototype our proposed system in a simulation environment with real world data produced by the Active BAT system [5].

This paper continues as follows: section 2 describes middleware architecture, section 3 discusses event filtering/aggregation/correlation, section 4 reports an experimental prototype with the Active BAT system, section 5 describes related works and it concludes with section 6.

2 Middleware Architecture

Service Oriented Architecture (SOA) is a well-proven concept for distributed computing environments. It decomposes applications, data, and middleware into

reusable services that can be flexibly combined in a loosely coupled manner. SOA maintains agents that act as software services performing well-defined operations. This paradigm allows users to focus on the operational description of the service. All services have an addressable interface and communication via standard protocols and data formats (i.e., messages). SOA can deal with aspects of heterogeneity, mobility and adaptation, and offers seamless integration of wired and wireless environments. Generic service elements are context model, trust and privacy, mobile data management, configuration, service discovery, event notification. The following are key issues addressed in our design.

- Support for service discovery mechanisms (e.g., new and sporadical services) for ad hoc networks.
- Support for an adaptive abstract communication model (i.e., event-based communication for asynchronous communication).

Peer-to-peer networks and grids offer promising paradigms for developing efficient distributed systems and applications. We integrate the Open Services Gateway Initiative (OSGi) [10] on the application layer. OSGi is open to almost any protocol, transport or device layers. The three key aspects of the OSGi mission are multiple services, wide area networks, and local networks and devices. Key benefits of the OSGi are platform and application independent. In other words, the OSGi specifies an open, independent technology, which can link diverse devices in local home network. The central component of the OSGi specification effort is the services gateway. The services gateway enables, consolidates, and manages voice, data, Internet, and multimedia communications to and from the home, office and other locations. We have developed a generic reference architecture applicable to any ubiquitous computing space. The middleware contains separate physical, sensor components, event-broker, service, service management, and an open application interface. We are in progress of implementing the reference architecture.

2.1 Service Semantics

We define service semantics in addition to the service definition so that services can be coordinated. Model real world is a collection of objects, where objects maintain their state using sensor data. Queries and subscriptions are examples of objects that are mapped to the service objects, and thus mapped to the sensors. This approach gives flexibility to services that will develop and evolve.

2.2 Layer Functionality

We describe the brief functionality of each layer below (see also Fig. 1).

Physical Layer: This layer consists of various sensors and actuators.

Sensor Component Layer: A sensor component layer can communicate with a wide variety of devices, sensors, actuators, and gateways and represent them

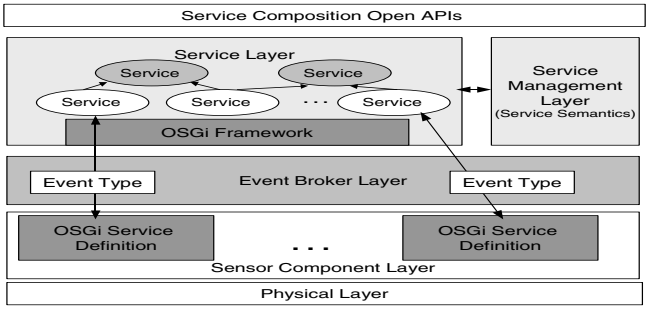


Fig. 1. Middleware Architecture with Wireless Sensor Data

to the rest of the middleware in a uniform way. A sensor component converts any sensor or actuator in the physical layer to a service that can be programmed or composed into other services. Users can thus define services without having to understand the physical world. Decoupling sensors and actuators from sensor platforms ensures openness and makes it possible to introduce new technology as it becomes available.

Event Broker Layer: This layer is a communication layer between Sensor component layer and Service layer. It supports asynchronous communication by publish/subscribe paradigm. Event filtering, aggregation, and the correlation service is currently a part of this layer.

Service Layer: This layer contains the Open Services Gateway Initiative (OSGi) framework, which maintains leases of activated services. Basic services represent the physical world through sensor platforms, which store service bundle definitions for any sensor or actuator represented in the OSGi framework. A sensor component registers itself with the service layer by sending its OSGi service definition. Application developers create composite services by Service Management Layer's functions to search existing services and using other services to compose new OSGi services. Canned services, which may be usefully globally, could create a standard library.

A context is represented as an OSGi service composition, where the context could be obtained. The context engine is responsible for detecting and managing states.

Service Management Layer: This layer contains ontology of the various services offered and the appliances and devices connected to the system. Service advertisement and discovery use service definitions and semantics to register or discover a service. Service definitions are tightly related to the event types used for communication in Event Broker Layer including composite formats. The reasoning engine determines whether certain composite services are available.

Application Interface: An application interface provides open interfaces for applications to manage services including managing contexts.

3 Event Correlation/Filtering/Aggregation

Event Correlation will be essential when the data is produced in a WSN and multi-step operation from event sources to final subscribers, which may reside in the Internet. Combined data collected by wireless devices into higher-level information or knowledge must be supported. In [12], we introduced a generic composite event semantics, which combines traditional event composition and a generic concept of data aggregation in wireless sensor networks. The main focus is on supporting time and space related issues such as temporal ordering, duplication handling, and interval-based semantics, especially for wireless network environments. We presented event correlation semantics defining precise and complex temporal relationships among correlated events.

Event correlation is sometimes deployed as part of applications, event notification services, or as a framework as part of middleware. Definition and detection of composite events especially over distributed environments vary, and equally named event composition operators do not necessarily have the same semantics, while similar semantics might be expressed using different operators. Moreover, the exact semantic description of these operators is rarely explained. Thus, we define the following schema to classify existing operators: *conjunction, disjunction, sequence, concurrency, negation, iteration, and selection.* Considering the analyzed systems, it becomes clear that to simply consider the operators is not sufficient to convey the full semantic meaning. Each system offers parameters, which further define/change the operator's semantics. The problem is that the majority of the system reflects parameters within the implementations.

Many event-based middleware offer a content-based filtering, which allows subscribers to use flexible querying languages to declare their interests with respect to event contents. The query can apply to different event types. On the other hand, the event correlation addresses the relation among event instances of different event types. Filtering and correlation share many properties. WSN has led to new issues to be addressed in event correlation. Traditional networking research has approached data aggregation as an application specific technique that can be used to reduce the network traffic.

WSN Data aggregation in-network operation has brought a new paradigm to summarize current sensor values in some or all of a sensor network. TinyDB [8] is an inquiry processing system for the sensor network and takes a data centric approach, where each node keeps the data, and those nodes execute retrieval and aggregation (in-network aggregation) with on-demand based operation to deliver the data to external applications. TinyLIME [3] is enhancing LIME (Linda In Mobile Environments) to operate on TinyOS. In TinyLIME, tuple space subdivided as well as LIME is maintained on each sensor node, and coordinated tuple space is formed when connecting with the base station within one hop. It works as middleware by offering this abstracted interface to the application. The current form of TinyLIME does not provide any data aggregation function, and only a data filtering function based on Linda/LIME operation is provided at the base station node. On the other hand, TinyDB supports data aggregation function via SQL query, but redundancy/duplication handling is not clear from available

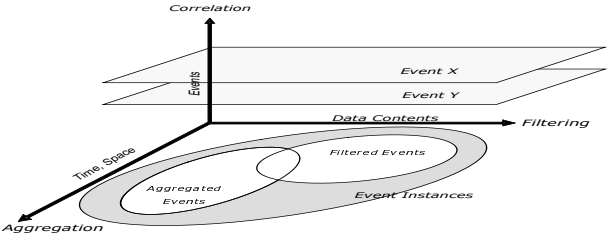


Fig. 2. Event Filtering, Aggregation and Correlation

documents. The coordination of nodes within WSN is different from other wireless ad hoc networks. The group of nodes acts as a single unit of processors in many cases, and for a single phenomenon, multiple events may be produced to avoid the loss of event instances, which is a different concept from traditional duplication of events. Aggregation has three stages: local, neighbour, and global. Fig. 2 highlights the relation among aggregation, filtering and correlation. Middleware research for WSN has been recently active, but most research focuses on in-network operation for specific applications. We provide a global view of event correlation over whole distributed systems from the correlation semantics point.

4 Prototype with Active BAT System

Sentient computing is a type of ubiquitous computing which uses sensors to perceive its environment and react accordingly. Sensors are used to construct a world model, which allows location-aware or context-aware applications. One research prototype of a sentient computing system has been the work at AT&T Research in the 1990s and the research continues at the University of Cambridge as the Active BAT system [5]. It is a low-power, wireless indoor location system accurate up to 3 cm. It uses an ultrasound time-of-light trilateration1 technique to provide more accurate physical positioning. Users and objects carry Active BAT tags. In response to a request that the controller sends via short-range radio, a BAT emits an ultrasonic pulse to a grid of ceiling-mounted receivers. At the same time the controller sends the radio frequency request packet, it also sends a synchronized reset signal to the ceiling sensors using a wired serial network. Each ceiling sensor measures the time interval from reset to ultrasonic pulse arrival and computes its distance from the BAT. The local controller then forwards the distance measurements to a central controller, which performs the trilateration computation. Statistical pruning eliminates erroneous sensor measurements caused by a ceiling sensor detecting a reflected ultrasound pulse, instead of one that travelled along the direct path from the BAT to the sensor.

SPIRIT (SPatially Indexed Resource Identification and Tracking) [5] provides a platform for maintaining spatial context based on raw location information derived from the Active BAT location system. It uses CORBA to access information and spatial indexing to deliver high-level events such as 'Alice has entered the kitchen' to listening context aware applications. SPIRIT models the

physical world in a bottom up manner, translating absolute location events for objects into relative location events, associating a set of spaces with a given object and calculating containment and overlap relationships among such spaces, by means of a scalable spatial indexing algorithm.

4.1 Prototype

The current Active BAT system employs a centralized architecture, and all data are gathered in the database, where computational power is cheap. The Active BAT system, as described, is expensive to implement in that it requires large installations, and a centralized structure. The centralized structure allows for easy computation and implementation, since all distance estimates can be quickly delegated to a place where computational power is cheap. Moreover, the active mobile architecture facilitates the collection of multiple simultaneous distance samples at the fixed nodes, which can produce more accurate position estimates relative to a passive mobile architecture.

It is inherently scalable both in terms of sensor data acquisition and management as well as software components. However, when constructing real-time mobile ad hoc communications with resource constrained devices, a distributed coordination must be supported, so that mobile device users can promptly subscribe to certain information.

We simulate all rooms and corridors hold gateway nodes (see the location map Fig. 3), which are capable to participate in event broker grids. The software design follows the service-oriented architecture described in Section 2. Thus, each local gateway node performs event filtering and correlation. Each local node registers the service that associates states with abstractions such as "ID10 in the room SN07". These states are decomposed to the units executable by the event broker grid, where event correlation and aggregation and filtering are operated. The details of high-level language for service composition and event type definition are still under development. The used data is taken on March 22nd in 2005. The total number of events is around 200,000, and a sample of event data is shown in Fig. 4. This shows BAT data after the location of the user is calculated, which consists of timestamp, user, area, coordination (X, Y, Z) and orientation. The receiver on the ceiling produces more than two times of data than this.

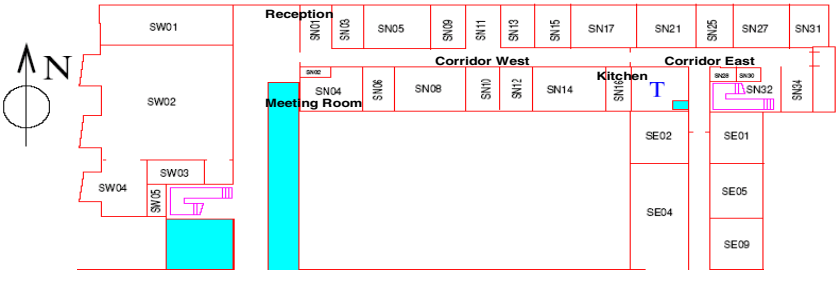


Fig. 3. Active BAT Location Map

```
30408.802618,10,SN09,1002.335327,1033.320801,1.261441,-22.443605
30408.856115,10,SN09,1002.520386,1033.289429,1.251856,-20.335649
30409.063099,10,SN09,1002.533203,1033.279297,1.285185,-20.326197
30409.112594,10,SN09,1002.732910,1033.234863,1.270585,-22.712467
30409.315079,10,SN09,1002.921448,1033.175903,1.271525,-54.598316
30409.370575,10,SN09,1002.994690,1033.126587,1.283121,-56.499645
30409.564561,10,SN09,1003.170227,1033.044556,1.298443,-52.581676
```

Fig. 4. Active BAT Events

4.2 Experiments

We performed several event correlations, and among those, we show the use of durable events below. Fig. 5 depicts the number of events over the local gateway nodes without durable events and Fig. 6 shows the same operation with durable event composition. During this experiment, thirteen BAT holders participated, which are shown ID1 through ID13. The result shows a dramatic reduction of event occurrences through the use of durable events.

Fig. 7 and Fig. 8 also depict the power of durable events composition over user ID 10 and 13 over the timeline (24 hours).

4.3 Temporal Ordering in Active BAT System

The applications derived from Active BAT have high accuracy and real-time tracking requirements. Problems of time synchronization and coordination amongst beacons are easily resolved, because these systems are wired and have a centralized controller. In the Active BAT system, the timestamp is derived from a Global Clock Generator (GCG), which is a hardware clock that sends 'ticks' to every component of the system over a serial link. When a location is computed, the location message is timestamped using the GCG. In general, GCG delay is in the order of microseconds, and the slowest part of the system is the bit that waits for ultrasound to propagate (speed of sound) after a position is requested but before a position can be calculated. This delay is used to measure

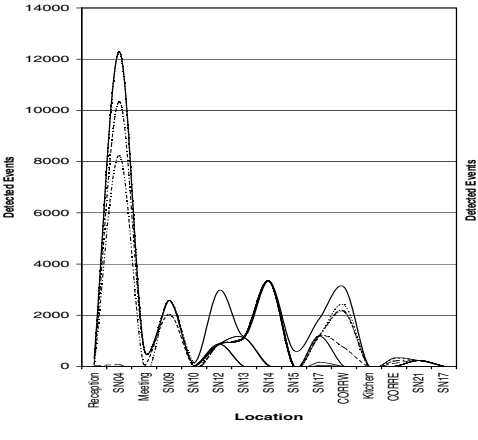


Fig. 5. All Sensed Events

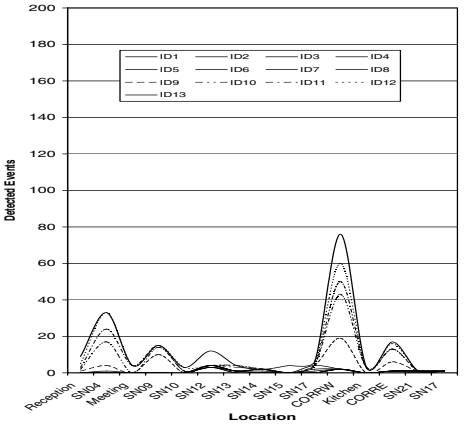


Fig. 6. Durable Events

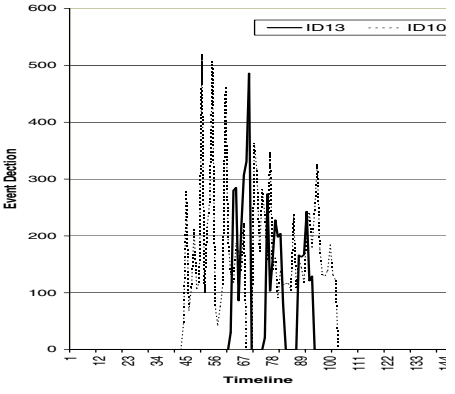


Fig. 7. Events of ID13 and ID10

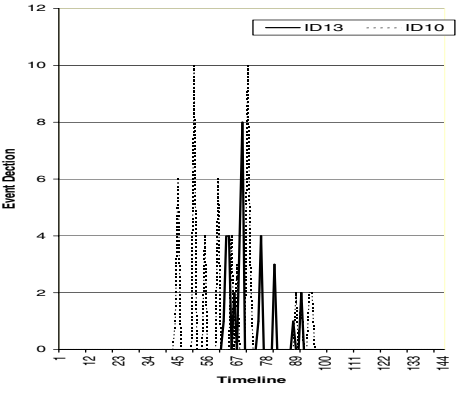


Fig. 8. Durable Events of ID13 and ID10

the distance between the BAT and the receiver at the ceiling. Once the location is calculated, the message then has to travel up to SPIRIT (order of milliseconds), and the event will be generated. However, no reliable information on that delay is considered. In Active BAT system, time synchronization is controlled in a centralized manner, and a trigger to collect BAT information is also used to synchronize all the clocks in the system. The current experiment assumes that all timestamps are properly synchronized. The implementation of temporal ordering mechanism described in [12] is in progress.

5 Related Works

Much composite event detection work has been done in active database research. SAMOS [4] uses Petri nets, in which event occurrences are associated with a number of parameter value pairs. Early language for composite event follows the Event-Condition-Action (ECA) model and resembles database query algebras with an expressive syntax. Snoop [2] is an event specification language for active databases, which informally defines event contexts. The transition from a centralized to a distributed system let to a new challenge to deal with time. Snoop presents an event-based model for specifying timing constraints to be monitored and to process both asynchronous and synchronous monitoring of real-time constraints. Reference [7] proposes an approach that uses occurrence time of various event instances for time constraint specification. GEM [9] allows additional conditions, including timing constraints to combine with event operators for composite event specification. In event-based middleware, publish/subscribe can provide subscription for the composite event instead of multiple primitive events, and this reduces the communication within the system and potentially gives a higher overall efficiency, which is addressed by [11]. Hayton et al. [6] on composite events in the Cambridge Event Architecture [1] describe an object-oriented system with an event algebra that is implemented by nested pushdown FSA to handle parameterized events. For related work on WSN data aggregation, see Section 3.

6 Conclusions and Future Work

The recent evolution of ubiquitous computing with a dramatic increase of event monitoring capabilities by wireless devices and sensors requires sophisticated middleware. We focus on specific aspects for supporting service broker grids including the data and communication models. The network environments will be more heterogeneous than ever, and open, P2P based networking environments will become common. In this paper, we provide a prototype of such a grid system over sensor networks and show simulated experimental results based on real data from the Active BAT system. Event broker nodes that offer data aggregation services can efficiently coordinate data flow. Event broker grids should seamlessly disseminate relevant information generated by deeply embedded controllers to all interested entities in the global network, regardless of specific network characteristics, leading to a solution to construct large distributed systems over dynamic hybrid network environments. We are working on a complete implementation including various timestamping environments and parallel/hierarchical composition.

Acknowledgment. This research is funded by EPSRC (Engineering and Physical Sciences Research Council) under grant GR/557303.

References

1. Bacon, J. et al. Generic Support for Distributed Applications. *IEEE Computer*, 68-77, 2000.
2. Chakravarthy, S. et al. Snoop: An expressive event specification language for active databases. *Data Knowledge Engineering*, 14(1), 1996.
3. Curino, C. et al. TinyLIME: Bridging Mobile and Sensor Networks through Middleware. *Proc. PerCom*, 2005.
4. Gatziu, S. et al. Detecting Composite Events in Active Database Systems Using Petri Nets. *Proc. RIDE-AIDS*, 1994.
5. Harter, A. et al. The Anatomy of a Context-Aware Application. *Proc. MobiCom*, 1999.
6. Hayton, R. et al. OASIS: An Open architecture for Secure Inter-working Services. *PhD thesis, Univ.of Cambridge*, 1996.
7. Liu, G. et al. A Unified Approach for Specifying Timing Constraints and Composite Events in Active Real-Time Database Systems. *Proc. IReal-Time Technology and Applications Symposium*, 1998.
8. Madden, S. et al. TAG: A tiny aggregation service for ad-hoc sensor networks. *Proc. of Operating Systems Design and Implementation*, 2002.
9. Mansouri-Samani, M. et al. GEM: A Generalized Event Monitoring Language for Distributed systems. *IEE/IOP/BCS Distributed systems Engineering Journal*, 4(2), 1997.
10. OSGi, http://www.osgi.org.
11. Pietzuch, P. Shand, B. and Bacon, J. Composite Event Detection as a Generic Middleware Extension. *IEEE Network Magazine, Special Issue on Middleware Technologies for Future Communication Networks*, 2004.
12. Yoneki, E. and Bacon, J. Unified Semantics of Event Correlation over Time and Space in Hybrid Network Environments. *Proc. CoopIS*, 2005.

Distributed Authentication in GRID5000

Sebastien Varrette[1,4], Sebastien Georget[2], Johan Montagnat[3], Jean-Louis Roch[4], and Franck Leprevost[1]

[1] University of Luxembourg, CESI-LACS, Luxembourg, INRIA DREAM team
[2] CNRS I3S unit, RAINBOW team
[3] Sophia Antipolis, France
[4] MOAIS/RAGTIME Project, ID-IMAG Laboratory, Grenoble, France

Abstract. Between high-performance clusters and grids appears an intermediate infrastructure called cluster grid that corresponds to the interconnection of clusters through the Internet. Cluster grids are not only dedicated to specific applications but should allow the users to execute programs of different natures. This kind of architecture also imposes additional constraints as the geographic extension raises availability and security issues. In this context, authentication is one of the key stone by providing access to the resources. Grid5000 is a french project based on a cluster grid topology. This article expounds and justifies the authentication system used in Grid5000. We first show the limits of classical approaches that are local files and NIS in such configurations. We then propose a scalable alternative based on the LDAP protocol allowing to meet the needs of cluster grids, either in terms of availability, security and performances. Finally, among the various applications that can be executed in the Grid5000 platform, we present μgrid, a minimal middleware used for medical data processing.

1 Introduction

This article is motivated by the need for a robust authentication system in the Grid5000 platform[1]. This French project aims at building an experimental Grid platform gathering at least 8 sites geographically distributed in France. The main purpose of this platform is to serve as an experimental testbed for research in *grid computing*. The researchers in each of the implied laboratories can deploy various applications on the grid, for instance for data analysis. Consequently, they will have to be able to authenticate on each of the connected nodes. The authentication system is therefore a key element in this project and more generally in the field of the cluster grid as it allows the allocation of resources. At least three constraints should be addressed by this system:

- *Availability*: the system should work even in case of punctual disconnections
- *Security*: privacy and integrity of the data should be assured.
- *Delegation*: each administrator of a site should be able to manage its own users.

[1] `http://www.grid5000.org`

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 314–326, 2005.
© Springer-Verlag Berlin Heidelberg 2005

This article is organized as follows: §2 precises the context by presenting a classification of computing grids and detailing the notion of cluster grid. This paper is mainly directed to Linux systems as they constitute major actors in the field of the grid computing[2]. Yet, our results can be extended to other systems. After introducing naming services (§2.2), a section will be dedicated to directories and the LDAP protocol (§3). A large part of this article is dedicated to experimentations (§4) where different configurations will be tested. The analysis of these experiments will lead to the proposition of an authentication architecture for the Grid5000 platform. Finally, §5 illustrates the running of the authentication system by presenting an application for medical data processing.

2 Context

Computer grids, as defined in [1], are distributed infrastructures that gather thousands of computers geographically scattered and interconnected through the Internet. This type of platform that used to appear in the 90's knew several evolutions and can be classified today in two main families:

1. The *"Desktop Grids"* [2] typically steal idle cycles of desktop PCs and workstations through the Internet to compute parts of a huge problem. Whereas this type of grid has been recently integrated in the general problematics of computing grids [3], we prefer going on with separating both architectures.
2. The *"Computing Grids"* rather gather one or more dedicated clusters. A *cluster* connects several computers through a local network in order to provide a coherent set able to deal with parallel computations, network load balancing, fault-tolerance... Each machine is a *node* of the cluster. Of course, each user of the grid has to authenticate on each node. This can be done either by a local copy of the user's credentials on the nodes or by using a Naming Service (such as NIS[3]) able to broadcast this information across the network. In the case of grids, the naming service has a strong imperative of scalability, as will be exhibited further.

To be totally exhaustive, an additional distinction in the concept of computing grids is proposed here. This classification is based on administrative heterogeneity (as illustrated in Fig. 1) and allows to subdivide computing grids in three categories:

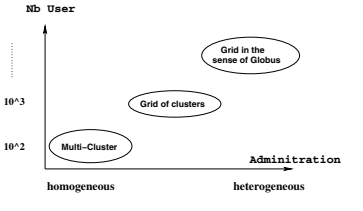


Fig. 1. Classification of computing grids based on administrative heterogeneity

1. Multi-clustering that bind together several nearby sites administrated by a single person that manages around 10^2 users.

[2] IBM which is the most represented manufacturers in the list of the 500 most powerful machines in the world (`http://www.top500.org`) claims 161 Linux clusters in this list. That is three quarters of the IBM systems and near a third of all manufacturers.

[3] Network Information System.

2. Cluster grid that merge several scattered sites through the Internet. Each site is administrated by different persons, yet the administrative domains are sufficiently open to enable the settlement of conventions between the sites (for instance when resolving the hostname of the nodes, or for the choice of a common authentication system). Such a topology, illustrated in Fig. 2, manages around 10^3 users and corresponds to the architecture of Grid5000.
3. Finally, the grid in the sense of Globus [4] manages a huge number of users in sites administratively closed. Traditional authentication solutions proposed in this context can also be applied for the first two cases but are unadapted for these topologies. This article proposes an authentication architecture adapted for the three cases.

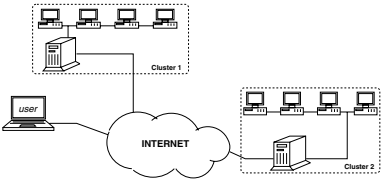


Fig. 2. A cluster's grid

2.1 Authentication of Users in Linux Environments

Under Linux, users authentication is based on two components:

1. The PAM system (Pluggable Authentication Module) which allows the transparent integration of various authentication technologies (as the UNIX standard based on `crypt`, RSA, DCE, LDAP) to be used by the services of the system as `ssh`, `passwd`, `su`, `ftp` ... PAM supplies an API by which the requests for authentication are mapped on specific actions related to the technologies used.
2. The NSS system (Name Service Switch). Once the user is authenticated, many applications still need to reach the information relative to this user. This information is traditionally contained in tables supplied either by local texts files (`/etc/passwd`, `/etc/shadow`, `/etc/group`, etc.) or by a naming service (see §2.2). NSS uses a common API dedicated to naming services that provided an intermediate layer between an application and a set of naming services. The NSS system is then able to access to the information of a given table (as `passwd` or `shadow`) using different naming services.

2.2 Naming Services

For a PC not connected to the Internet, user authentication is achieved through local tables (`passwd` and eventually `shadow`) stored in files. Similarly, the name resolution of hostnames into IP addresses uses the file `hosts`. For a cluster, administrators prefers to use a Naming Service[4] able to broadcast authentication

[4] DNS (Domain Name Service), NIS and NIS+ for instance.

information across the network to authorized nodes. The information is centralized on one or more servers, making the administrative task easier: without such service, every single node should maintain its own copy of the information.

NIS. Introduced by Sun in 1985, the Network Information Service is used to centralized administration of systems information. The information is stored in maps under indexed databases (db, dbm) reachable by RPC[5]. NIS uses a Master/Slave model but does not allow the treatment of important volumes of data (each modification involves the transfer of the totality of the base). Furthermore, it is particularly hard to organize the data in a hierarchical way and the access security remains weak. In spite of all these drawbacks, NIS remains a well used system at the level of clusters and local networks mainly because of its installation simplicity.

NIS+. It was the answer of SUN to the drawbacks of NIS.NIS+ introduces the distribution of the data between master and slave in an incremental way, by adding the notion of hierarchical tree for the data. The use of certificates solves the security issue. Yet, the luck of flexibility in the hierarchical structure together with a too complicated installation proceeding slow down the passage from NIS to NIS+. Nevertheless, this approach prefigures the concepts used in this article.

NetWare NDS. Among the various proprietary solutions available, NetWare[5] is a local-area network (LAN) operating system that runs on a variety of LANs.It provides users and programmers with a consistent interface that is independent of the actual hardware used to transmit messages. In particular, NetWare NDS is a version of the NetWare file server operating system. NDS stands for Netware Directory Services, and is a hierarchical directory used to manage user-IDs, groups, computer addresses, printers and other network objects in a convenient manner. NDS is then used to retrieve the information required by an authentication process. This approach is finally closed to the one presented in this paper but has the inconvenient of a prohibitive cost unadapted for our context.

3 Directories and LDAP

A directory is like a database, but tends to contain more descriptive information. Directories are tuned to give quick-response to high-volume lookup or search operations. They may have the ability to replicate information widely in order to increase availability and reliability, while reducing response time.

Based on X.500 protocol (ISO norm for the management of electronic directories), LDAP [6] is a standard access protocol to electronic directories allowing to perform researches and modifications. LDAP is based on the model of DNS: data are naturally organized in a tree structure and each branch of the tree can easily be distributed among different servers. LDAP technology has been

[5] Remote Procedure Call.

adopted by many large companies. The generalization of LDAP has also been implemented in the applicative bases (some operating systems, like Mac OS X or Solaris 9, integrate a LDAP directory). LDAP proposes mechanisms to manage authentication, authorization and confidentiality of the exchanges. Indeed, several methods of authentication corresponding to various security levels are available (login/password, login/password on SSL, X.509 certificate etc...). The possibilities of authentication can be extended with the SASL[6] API allowing to easily integrating new mechanisms of authentication like Kerberos The applications thus delegate the step of authentication to the LDAP directory which implements one of the quoted methods. To conclude in terms of security, LDAP provides various guarantees thanks to the integration of cypher and authentication standard mechanisms (SSL/TLS, SASL) coupled with Access Control Lists. All these mechanisms enable an efficient protection of transactions and access to the data incorporated in the LDAP directory. Thereafter, practical experimentations on LDAP were done through the open-source and reliable implementation OpenLDAP[7].

Organization of Data in LDAP. LDAP data are organized in a tree structure called Directory Information Tree (DIT). Each node of the tree corresponds to an entry of the directory. An example of DIT is provided in Fig 3. Each entry is

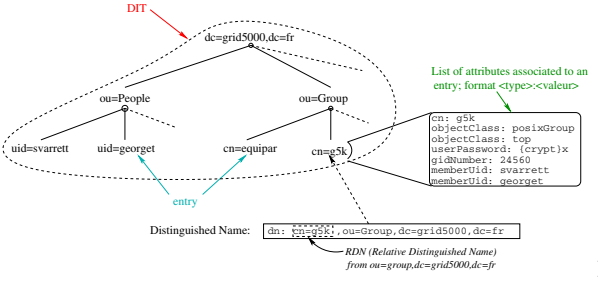


Fig. 3. Example of DIT: Case of the users management

DC	domain components
OU	organizational unit
O	organization
CN	common name
SN	surname
UID	user ID

Fig. 4. Main abbreviations used in the DN field

referred in a unique way by its *distinguished name (DN)*. This unicity is obtained by combination of the attributes listed in tab 4. The directory service provided by LDAP is based on a client/server model. The servers can be organized in the following configurations:

1. *Local Directory Service*: A unique server is able to deal with the clients requests.
2. *Local Directory Service with Referrals*: The server is configured to provide directory services for a local domain and to return referrals (i.e. a pointer) to a superior service capable of handling requests outside the local domain.

[6] Simple Authentication and Security Layer.
[7] `http://www.openldap.org`

3. *Replicated Directory Service*: Partial replication can be operated between master and slave servers.
4. *Distributed Local Directory Service*: The database is divided in subparts (eventually replicated) that are accessible through a set of referrals between the servers.

The last two modes will be particularly interested in our context.

LDAP vs Databases. LDAP is often compared to a database. It is globally the case even if differences exist: see tab 1.

Table 1. Advantages/Drawbacks of LDAP on Databases

Criteria	**LDAP**	**Databases**
R/W ratio	read optimized	R/W
scalability	easy (LDAP schema)	hard
Table distribution	inherent	rare [7]
Replication	possible	possible
Transactional model	simple	advanced
Standard	yes	no (specific to SGBD)

LDAP vs. NIS. This article compares authentication solutions based on LDAP to NIS. Tab 2 briefly introduce the characteristics of every system.

Table 2. Advantages/Drawbacks of LDAP on NIS

Criteria	**LDAP**	**NIS**
Port	specific (389/636 by default)	arbitrary (RPC)
Data privacy	possible	impossible
Access control mechanisms	yes	no
Table distribution	yes	no
Replication	yes (partial replication available)	yes (total repl. only)
Researches Semantics	advanced	simple

4 Experimentations

The comparison of authentication systems was realized on the client side by computing the number of simultaneous authentications the server can handle. So, the measures take into account the three elements of the identification chain: the PAM module, the transport layer between client and server and the delay in server's response. This approach allows to obtain results which correspond to the reality and not to the theoretical performances that the servers are supposed to achieve.

4.1 Local Model

While being hard to maintain, the local duplication on each nodes of the system tables `passwd`, `shadow` and `group` is not the most effective solution (see fig.5).

During the authentication process, the files are sequentially read. The authentication time directly depends on the number of entries in the table. This is naturally unadapted to the case of clusters and grids (with high number of users). We also show the influence of preloading the libraries involved.

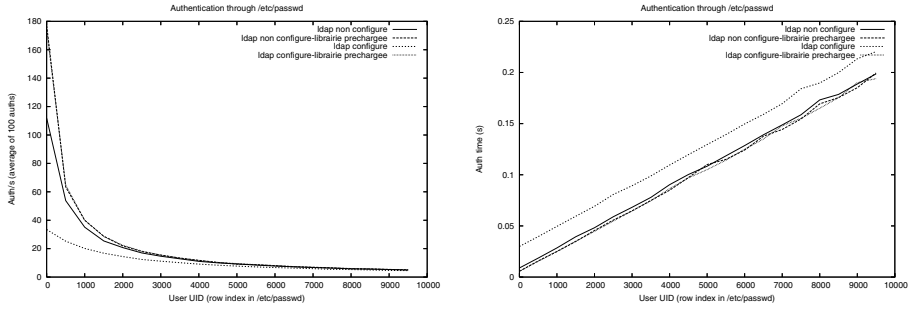


Fig. 5. Local table model: Impact of the location in the table on authentication time

4.2 Comparison Between NIS and the Local Model

Contrary to the local model, NIS ensures that the authentication time is globally independent from the number of entries in the table (see fig. 6). We know experiment different configuration of LDAP to justify our proposition for Grid5000.

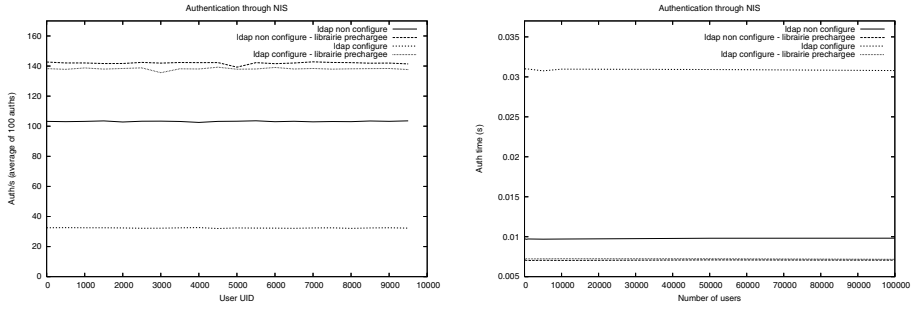


Fig. 6. In NIS, the authentication time is independent from the base size

4.3 Centralized Client/Server Model

In this model, the LDAP server is configured in a similar way to a NIS server: the contents of all the tables are on the same server reachable by every nodes of the grid. Fig.7 illustrates the structure of the table in the LDAP server. Before comparing this solution with NIS, we wanted to estimate the impact of the configuration of the LDAP server LDAP on its performances. This is done in the following sections.

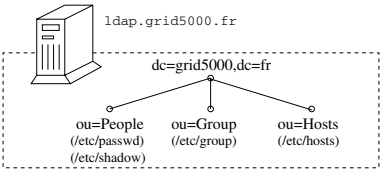


Fig. 7. Centralized client/server model

Impact of Data Indexing in LDAP Configuration. When installing a LDAP server, a basic indexing is proposed but fig.8 shows that it *should not be*

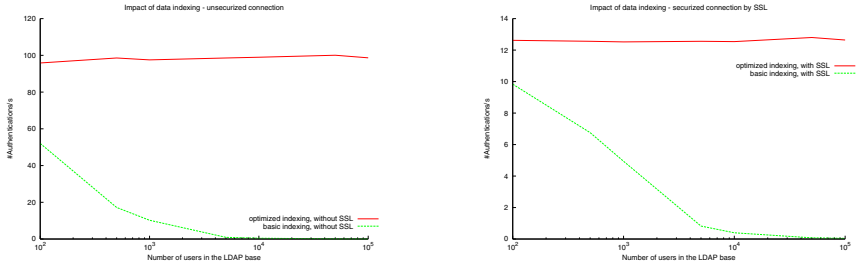


Fig. 8. Impact of data indexing on the LDAP server performances

used. An appropriate indexing[8] should be preferred. This configuration guarantees constant performances with regards of the number of entries in LDAP base. Using SSL divides the performances by 9. This will be confirmed in §4.3. This is required for resources authentications and communication privacy.

Impact of the Log Level. The LDAP server can log different information represented by a level (see fig.9). They can be combined by addition. Fig.10 shows that each level has a different impact on the server performances.

Niv.	Description	Impact
-1	enable all debugging	++++
0	no debugging	–
1	trace function calls	+++
2	debug packet handling	0
4	heavy trace debugging	+
8	connection management	++
16	print out packets sent/received	0
32	search filter processing	+
64	configuration file processing	0
128	access control list processing	+++
256	stats log connections/op/results	+
512	stats log entries sent	≃0
1024	print coms with shell backends	0
2048	print entry parsing debugging	0

Fig. 9. Log levels in OpenLDAP

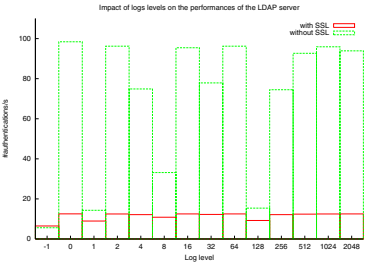


Fig. 10. Impact of logs levels on performances

Impact of the Backend Used. OpenLDAP supports a variety of database backends which you can use ("`ldbm`" by default). The first measures made to compare the most used backends does not reflect real differences in the server performances.

Impact of SSL on LDAP. As seen in §2.1, user authentication uses two components: PAM and NSS library. In LDAP, a specific interface must be used for each of these libraries. It can be configured to use SSL or not in the communications between the client and the server. As mentioned in §4.3, SSL divides the performances by 9 (see tab.3). This influence can be explained when ana-

[8] See `http://www.openldap.org/faq/data/cache/42.html` for instance.

Table 3. Impact of using SSL in PAM and NSS libraries on the performances - intra-cluster Measures

PAM		NSS		#authentications/s		
ldap	ldaps	ldap	ldaps	Interval	Average	with preload
X		X		between 94 and 97	95.99	145.9
X			X	between 92 and 95	94.15	144.4
	X	X		between 12 and 13	12.46	14.28
	X		X	between 12 and 13	12.51	14.27

lyzing the number of messages exchanged during an authentication by `ldap` or by `ldaps`. When using the `ldap` protocol, 45 LDAP messages are exchanged together with 35 TCP messages. With `ldaps`, it is 70 TLS messages and 47 TCP messages. Communications are thus more important, and encryption/decryption time should be taken into account.

Impact of Inter-Cluster Latency. The measures presented in the tab.4 were made with a server located in Sophia Antipolis while clients belongs to the cluster of Grenoble. Performances are divided by 4 because of latency and

Table 4. Impact of inter-cluster latency

PAM		NSS		#authentications/s	
ldap	ldaps	ldap	ldaps	interval	average
X		X		from 15 to 22	20.1
X			X	from 7 to 23	17.6
	X	X		from 6 to 7	6.89
	X		X	from 5 to 7	6.79

network disturbances. Consequently, the inter-clusters communications should be minimized in the authentication process.

We realized a similar analyse with a NIS server. Results are displayed in tab.5 As communications are less important in NIS, the performances are globally

Table 5. Impact of latency when using a NIS server

Latency type	#authentications/s			
	interval	average	with preload	average with preload
Intra-Cluster	from 230 to 310	263.0	from 266 to 338	290
Inter-Cluster	from 32 to 37	35.1	from 31 to 38	35.3

better with it. Yet, the advantage of LDAP comes from its capacity to distribute the tables with eventually a partial duplication. This configuration is presented in the following section.

4.4 Distributed "flat" Model

Based on a DNS model, LDAP allows the distribution of the tables on multiple servers. As before, the tree structure used to store the data in the LDAP server follows the organization of the sites in the grid but here, each site is responsible for a sub-tree containing data relative to the users and the resources of the site as illustrated in fig. 11. Reaching the data contained in other branches can be done in two ways:

- by using referrals as a pointer to a server able to answer a request
- by partial replication of some or all the other branch's.

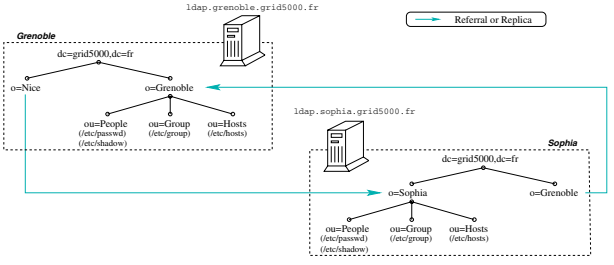


Fig. 11. flat distributed model for authentication based on LDAP

We propose a partial replicated approach based on the following organization :

- each site is master for its branch and slave for the others
- A special backend (`meta`) aggregates each branch into the LDAP bases.

Table 6. Experiments for a partial replicated approach between `o=grenoble` and `o=sophia`

	backend `meta`		#auth./s (`ldap`)			#auth./s (`ldaps`)			
Mode	Grenoble	Sophia	min.	avg.	max.	min.	avg.	max.	Comments
centralized	inact.	inact.	84.9	89.0	107.3	16.7	17.4	17.9	Centralized mode (see §4.3)
replica	inact.	inact.	82.9	96.6	108.9	17.2	17.4	17.8	Impact of replica configuration
replica	local	local	82.0	87.9	97.1	16.9	17.1	17.5	User managed in Grenoble
replica	local	local	81.3	88.9	97.6	16.8	17.2	17.5	User managed in Sophia
replica	local	remote	35.8	36.8	38.3	13.3	13.4	13.7	User managed in Grenoble impact of remote backend
replica	local	remote	16.1	16.5	17.0	9.2	9.2	9.4	User managed in Sophia impact of remote backend

Our experimental results for this architecture are presented in tab.6. It can be seen that this approach (with local backends) guarantees the performances with regard to the centralized model (see §4.3). In addition, contrary to the referrals approach, this architecture is able to solve the problem of availability

(if a cluster is disconnected, the authentication system is still running). Security is ensured by the protocol LDAP itself whereas delegation is due to the tables' distribution in proposed architecture. This system is therefore a particularly good candidate for a robust authentication system in a distributed environment such as Grid5000.

5 μgrid and Application to Medical Data Processing

Grid5000 is an experimental platform for grid computing research that is not making any assumption on the middleware to be used. Instead, Grid5000 users are deploying the middleware they need for their research and experiments. We have deployed the μgrid middleware [8] over the Grid5000 infrastructure. μgrid is a lightweight middleware prototype that was developed for research purposes and already used to deploy applications to medical image processing [9].

The μgrid middleware was designed to use clusters of PCs available in laboratories or hospitals. It is intended to remain easy to install, use and maintain. Therefore, it does not make any assumption on the network and the operating system except that independent hosts with private CPU, memory, and disk resources are connected through an IP network and can exchange messages via communication ports. This matches the Grid5000 platform. The middleware provides the basic functionalities needed for batch-oriented applications: it enables transparent access to data for jobs executed from a user interface. The code of μgrid is licensed under the GPL and is freely available from the authors web page.

The application considered in our experiments is an application to medical image database analysis that is further detailed in [9]. The objective is too face the huge amount of medical data that one can store on a grid by providing a medical image search tool. Medical images are stored together with accompanying metadata (information on patients, acquisition devices, hospitals, etc). The structure of medical data and metadata is often complex and there are very strong privacy constraints applying on both of them: only a very limited number of authorized people should be able to access medical data content. This is even more critical on a grid infrastructure given the data dispersion and the number of users with a potential access to the data.

To retrieve a medical data, a selection is first done on the associated metadata. This enables queries such as "find the MR Image of Mr Patient, acquired yesterday in this hospital". However, there are many clinical cases where a physician would like to be able to find relevant and similar medical cases to an image he is studying to confirm his/her diagnosis. Given the tremendous amount of medical images produced daily, it is impossible for the user to manually browse through the whole medical image database. An hybrid request is need insteadto perform that kind of query: first some potential candidate images are selected using a query on metadata, and then the candidates are compared to the sample image through a compute intensive image analysis step. A grid is well adapted to handle the computation involved as the images may be distributed over the

grid nodes for parallel computations. The kind of image analysis to apply is very dependent on the clinical domain and the features of interests. In [9], we used simple similarity measurements algorithms that give relevant results when looking for visually similar images. Thanks to the gridification of this application, very significant speed-up can be achieved in using a grid infrastructure (it highly depends on the amount of resources available). Thanks to a strict authentication procedure, it is possible to identify users and to check whether they are authorized to access the data to ensure medical privacy.

6 Conclusion

The experiments done confirmed the advantages of the NIS system as a fast deployed and efficient solution. It does perfectly fit the requirements of a cluster where security constraints are weak. Yet, the grid context and more precisely the Grid5000 platform places those constraints in the foreground. The geographic distance which separates the sites does not allow to use a dedicated and controlled network. It is then necessary to ensure the confidentiality of the exchanged information at the level of the used protocols. LDAP, and more exactly the `ldaps` protocol, supplies this feature. Passing from clusters to grids opens also organization issues between distinct administrative entities. In that case, a centralized model (either based on NIS or LDAP) no longer applies: administrators of each site want to manage their own users. Delegation can be obtained by several ways (see [10,3] for instance). Here, we split the LDAP namespace across multiple servers and arrange LDAP servers in a hierarchy following the administrative domains (see fig 11). Each site hosts a master server for its relative branch. Then, reaching the information contained in other servers can be done either by the "referrals" mechanism of by replication. Both solutions are possible, but the grid context raises the availability issue too. High-availability is especially critical for enterprise and grids authentication services, because in many cases the system will come to a stop when authentication stops working. That's why a solution based on a partial replication is proposed: in a normal configuration, a referral on a disconnected server jams the authentication of a user handled by this server. This is not the case with a local replication. To sum up, the proposed infrastructure supports heterogeneity, compensates the security flaws of NIS and solves the security, availability and delegation constraints required in the Grid5000 platform which adopted this system.

Evolutions are already planned with the integration of additional information in the LDAP directory such as installed softwares and cluster configurations. The objective is to create a repository used by grid services such as DNS, the monitoring and discovery service or the batch scheduler. We are also looking toward the evaluation of a referral based solution using the OpenLDAP proxy cache[11]. A comparison to Globus and more precisely GSI is also planned.

Finally, the authors want to thank Olivier Richard, Nicolas Capit and Julien Leduc from the ID-IMAG Laboratory for their technical contribution.

References

1. Foster, I., Kesselman, C.: Globus: A metacomputing infrastructure toolkit. International J. of Supercomputer Applications and High Performance Computing **11** (1997) 115–128
2. Fedak, G., Germain, C., N'eri, V., Cappello, F.: Xtremweb: A generic global computing system. In: In IEEE Int. Symp. on Cluster Computing and the Grid. (2001)
3. Foster, I.: The anatomy of the Grid: Enabling scalable virtual organizations. Lecture Notes in Computer Science **2150** (2001)
4. Foster, I., Kesselman, C., Tsudik, G., Tuecke, S.: A Security Architecture for Computational Grids. In: Fifth ACM Conference on Computer and Communications Security Conference, San Francisco, California (1998) 83–92
5. Nowell Corporation: Netware 6 (2005) `http://www.novell.com/documentation/nw6p/index.html`.
6. Wahl, M., Howes, T., Kille, S.: RFC 2251 - Lightweight Directory Access Protocol (v3). Technical report, IETF (1997) `http://www.ietf.org/rfc/rfc2251.txt`
7. Stonebraker, M., Aoki, P.M., Devine, R., Litwin, W., Olson, M.A.: Mariposa: A new architecture for distributed data. In: International Conference on Data Engineering (ICDE). (1994) 54–65
8. Seitz, L., Montagnat, J., Pierson, J.M., Oriol, D., Lingrand, D.: Authentication and autorisation prototype on the microgrid for medical data management. In: Healthgrid'05, Oxford, UK (2005)
9. Montagnat, J., Breton, V., Magnin, I.: Partitionning medical image databases for content-based queries on a grid. Methods of Information in Medicine **44** (2005)
10. Varrette, S., Roch, J.L., Denneulin, Y., Leprevost, F.: Secure Architecture for Clusters and Grids. In IEEE, ed.: Proceedings of the 2ème Conférence Internationale sur les Infrastructures Critiques CRIS 2004), Grenoble, France (2004)
11. Apurva, K.: The OpenLDAP Proxy Cache. Technical report, IBM Research lab of India (2003)

A Load Balance Methodology for Highly Compute-Intensive Applications on Grids Based on Computational Modeling

D.R. Martínez, J.L. Albín, J.C. Cabaleiro, T.F. Pena, and F.F. Rivera

Dept. Electronics and Computing, Univ. Santiago de Compostela, Spain

Abstract. Compute-intensive simulations are currently good candidates for being executed on distributed computers and Grids, in particular for applications with a large number of input data whose values change throughout the simulation time and where the communications are not a critical factor. Although the number of computations usually depends on the bulk of input data, there are applications in which the computational load depends on the particular values of some input data. We propose a general methodology to deal with the problem of improving load balance in these cases. It is divided into two main stages. The first one is an exhaustive study of the parallel code structure, using performance tools, with the aim of establishing a relationship between the values of the input data and the computational effort. The next stage uses this information and provides a mechanism to distribute the load of any particular simulating situation among the computational nodes. A load balancing strategy for the particular case of STEM-II, a compute-intensive application that simulates the behavior of pollutant factors in the air, has been developed, obtaining an important improvement in execution time.

1 Introduction

Grid environments have become an alternative to the use of traditional supercomputers in parallel compute-intensive applications. In particular, it is true for numerical simulations. Pools of servers, storage systems and networks in a large virtual computer system can be used in a Grid environment. However, an optimal load balancing strategy is critical in a Grid environment in order to avoid processing delays and overcommitment of resources. On distributed heterogeneous behaviors, like Grids, load balancing is a great challenge because the power of each computational node must be taken into account and it changes dynamically. Hence, load balancing for these systems is much more complex than in other environments, and good performance predictions of application components is absolutely essential to success [1].

Our purpose is to introduce a methodology to obtain an optimized load balancing distribution for highly compute-intensive applications in which information from the input data values affect the performance of the parallel code. A methodology based on two main stages is proposed. The first one is an analysis

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 327–336, 2005.
© Springer-Verlag Berlin Heidelberg 2005

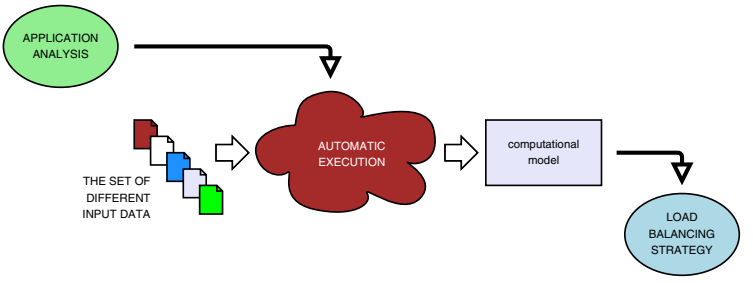


Fig. 1. Methodology scheme

of the application to produce a computational model of the application based on input data values. This model is obtained by exhaustive executions of the application using the complete spectrum of input data. This model must supply a prediction about how the computational load is influenced by the application space. In the second stage, a load balancing strategy based on the models obtained in the previous analysis is performed. Fig. 1 shows a scheme of this methodology.

The methodology we propose is focused on compute-intensive applications based on a discrete mesh of the simulated space, where communications are not a critical factor and the computational load depends on values of input data. Applications like 3D simulation of HEMT semiconductor devices [2] or finite-difference simulation of physical processes, like STEM-II [3], fit into the applications in which the methodology can be applied. A fine tuning in the performance of these kinds of applications is critical, especially in Grid environments, because of their high time-consuming and their intensive use of resources.

In this paper, the proposed methodology has been used in order to optimize the load balance in STEM-II [3]. STEM-II is a Eulerian numerical model that accounts for the transport, chemical transformation, and removal of atmospheric pollutants. Due to its compute-intensive attribute, an optimized distribution of computational load among the processors is critical to obtain high efficiency and, for the end user, real time predictions. Meteorological input data, that change during the simulating time, have an important impact on the computational load. STEM II has been chosen as one of the case studies of the European CrossGrid project [4] for the development of new Grid technologies.

2 Modeling the Computational Behavior

Obtaining a computational model of the application is the first stage of the proposed methodology. The aim is to define a model by means of analytical expressions that establish the relationship between the distribution of the computational effort throughout the simulating space and the input data values.

Consider a simulation application based on a discrete mesh of the simulated space. The first step is to perform an analysis of the application code in order to obtain a set of parameters (P_i) that controls the main execution flow of each of

the nodes of the simulating mesh. This step is closely related to the application, so there is no general procedure to perform this analysis. In many applications these parameters could be those that control branches or the number of iterations in loops that have an important role in the computational cost of each simulating node. The next step is to define a mechanism that calculates, without executing the application itself, the values of the P_i parameters related to each node corresponding to specific input data values. Therefore, the behavior of each node of the simulating mesh corresponding to a specific simulated situation is known before the execution of the application.

In order to know the computational cost throughout the simulating space, it is necessary to obtain for each node the relationship between the combination of parameters and its associated performance. This can be performed using a wide range of valid different input data and measuring both the computational effort and the P_i parameters related to each node of the simulating mesh. The use of performance tools like PAPI [6] can help to obtain information about the computational effort, such as the number of instructions or floating point operations (FLOPs). These measurements can be performed in an automatic and parametric way. A relationship between a combination of parameters and its associated cost can be obtained from the results of these measurements.

Therefore, by calculating the associated workload to each node of the simulating mesh from the input data values, an approximation of how the computational effort is distributed throughout the simulation mesh, in a specific situation, is obtained without actually executing the application itself. A load balancing strategy can make use of this information. In Grid environments, a resource broker or a prediction tool, like PPC [5], can also take it into account to achieve good resource distributions and performance information.

3 Load Balancing Strategy

In homogeneous systems the distribution of the computational effort among the processors is a straightforward issue once the computational model of the application provides a map of how the workload is distributed throughout the simulating mesh. In heterogeneous environments, the information from the computational model should be joined to information about the computational power of the different processors in order to provide an efficient distribution of the workload. A good distribution can be performed in several ways. However, the most appropriate distribution depends on both the specific application and the underlying hardware.

Let us consider a n-dimensional simulating mesh and a heterogeneous n-dimensional processor mesh. The computational model provides the computational cost of each of the mesh nodes. Let $L(x_1, x_2, \ldots, x_n)$ be the computational load associated with the $(x_1, x_2, \ldots, x_n)$ coordinates in the simulating mesh, and $P(y_1, y_2, \ldots, y_n)$ the computational power associated with the $(y_1, y_2, \ldots, y_n)$ coordinates in the processor mesh. Note that the dimension of L and P are the same, in such a way that each dimension of the simulating mesh is associated

with one dimension of the processor mesh. In the first stage, a balanced distribution of the computational load along the first dimension is performed. The computational power of each set of processor sharing the same y_1 index is managed as a whole. In the same way, the computational load of simulating nodes with the same x_1 index is managed as a whole. Therefore, the sets of simulating nodes are distributed among the sets of processors so that all sets of processors have approximately the same computational load associated. Note that the computational power of the processor sets are taken into account. Therefore, groups that associate nodes and processors are obtained. These groups are independently managed in the next step in which, by means of the same strategy, the distribution on the next dimension (x_2 and y_2) is performed and so on. The final distribution is obtained after n steps.

As an example, consider, in detail, the 2-dimensional mesh case in which the computational power is given by the number of FLOPs. Let us consider a $C \times D$ processor mesh, where C and D are the sizes of y_1 and y_2 dimensions, respectively. Let $L(i, j)$ be the estimated FLOPs in the (i, j) coordinate of the $N \times M$ 2D simulating mesh. In the same way, N and M are the sizes of x_1 and x_2 dimensions, respectively. Then, the total FLOPs will be:

$$L = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} L(i, j) \tag{1}$$

Besides, let $L^{p,q}$ be the FLOPs of mesh nodes between the columns p and q:

$$L^{p,q} = \sum_{i=p}^{q-1} \sum_{j=0}^{M-1} L(i, j) \tag{2}$$

The idea of the load balancing strategy is to distribute the N columns of simulating mesh between the C columns of the processor mesh, so that the columns of the simulating mesh from i_s to i_{s+1}-1 are associated with the s-th column of processor mesh ($0 \leq s < C$) where i_s is obtained by:

$$L^{i_s, i_{s+1}} + \frac{L^{i_{s+1}-1, i_{s+1}}}{2} < L \times \frac{P_s}{P_\mathrm{T}} \leq L^{i_s, i_{s+1}} + \frac{L^{i_{s+1}, i_{s+1}+1}}{2} \tag{3}$$

where $i_0 = 0$. P_s and P_T represent the computational power, measured as FLOPs/sec, of the s-th column of processor mesh and the total computational power, respectively. If all the processors have the same computational power, P_s/P_T reduces to $1/C$.

In a second stage, in an independent way, each of the generated partitions is balanced, using the same strategy, among the D processors associated with the corresponding column of processor mesh. Fig. 2 shows an example of the distribution of a 8×8 simulating mesh among a 3×4 processor mesh, having all processors the same computational power. The numbers represent the computational load of each node of simulating mesh.

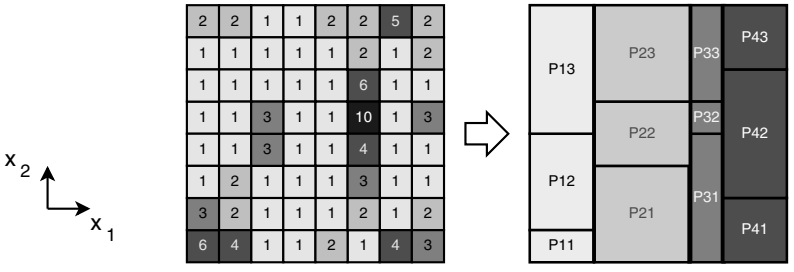


Fig. 2. Example of the load balancing strategy in a 2D simulating mesh

4 Case Study: STEM-II

STEM-II (Sulphur Transport Eulerian Model 2) [3] is a Eulerian air quality model which simulates transport, chemical transformations, emissions and depositions processes in an integrated framework. It can be used to know in advance how the meteorological conditions (obtained from a meteorological prediction model) affect the emission behaviors of a thermal power plant. The speedup in the simulation process is of great importance for saving time when decisions about modifications in the industrial process have to be made.

The model is computational intensive because the governing equations are nonlinear, highly coupled and stiff. As in other computational intensive problems, the ability to fully utilize these models remains severely limited by today's computer technology; thus, Grid computing should be applied to achieve a reasonable response time [7]. The `vertlq` module is the most costly part of the whole program. Inside the `vertlq` module there are two nested loops that go through the horizontal dimensions of simulating space. In each iteration of the inner loop, the `rxn` routine is executed. This routine is the most time-consuming phase, around 80% of the execution time of STEM-II program, this rate being highly dependent on the input data values. The greatest computational effort is reached by simulating meteorological situations with high water concentrations in the atmosphere.

The Grid-enable version of the STEM-II was developed jointly by groups of the Department of Electronics and Systems of the University of A Coruña, and the Department of Electronics and Computer Science of the University of Santiago de Compostela to be used in As Pontes Power Plant (A Coruña, Spain). It is focused on the parallelization of the horizontal spatial loops of the `vertlq` module [8]. Therefore, only the two horizontal dimensions of the simulating mesh are considered for the distribution of computational load. A block distribution of the simulating mesh is performed as default and no balancing strategy is taken into account at all. This can be inefficient in situations where the computational load is not distributed homogeneously throughout the simulating mesh. Due to the wide simulating area ($61 \times 61\,\mathrm{km}^2$) and its particular meteorological behavior, it is common to simulate situations with a high meteorological variation inside the

simulating area that often generates an inefficient load balancing and, therefore, a low performance.

4.1 Modeling the Computational Behavior of STEM-II

For modeling the computation behavior of STEM-II, an exhaustive analysis of the `rxn` routine was performed. An important conclusion of this analysis is that the variables which control the main loops and branches only depend on the values of some input data values related with the meteorology, specially with the water concentration. Therefore if the values of these variables are known, the main execution flow of the `rxn` routine can be predicted. Fig. 3 shows a simplified scheme of the `rxn` routine where the most relevant subroutines of `rxn` are shown, besides the loops and branches that have an important role in the

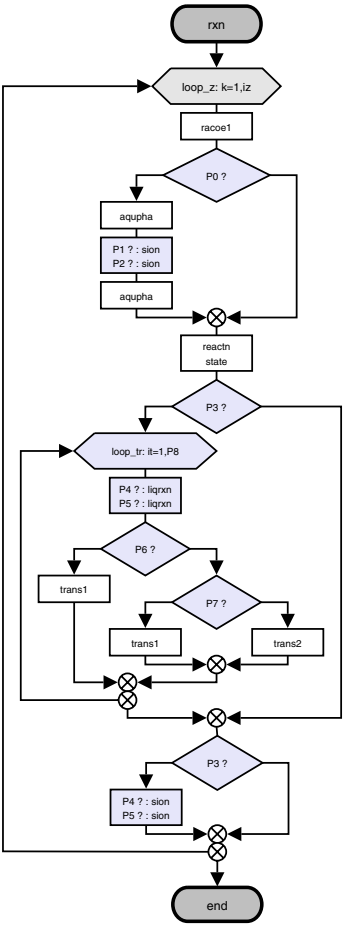


Fig. 3. Scheme of the `rxn` routine. The "`COND ? : name`" expression means the "`name`" subroutine is executed when "`COND`" is true.

behavior of the routine. The nine parameters ($P_i, 0 \leq i \leq 8$) that control the principal behavior of this routine are highlighted. For branches, these parameters can take two values: 1 if the branch condition is true or 0 if it is false. For loops, the parameter means the size of the loop. Although the `loop_z` loop is of great importance because it goes through the vertical spatial dimension of the simulating space, it has a fixed number of iterations. Therefore it is not necessary to take this loop into account to predict the `rxn` behavior. Then, each combination of the P_i parameters corresponds to a possible execution flow of the `rxn` routine. Since the parameters only depend on meteorological data, these parameters can be calculated, once meteorological input data values are known, in an easy and fast way, just by emulating the execution of the program. These operations have a negligible cost in comparison with the cost of the whole routine.

Once the parameters P_i are identified, it is necessary to evaluate the real influence of those parameters on the computational cost of the `rxn` routine. Using a wide range of different and real meteorological situations, the STEM-II program was executed. For each iteration of the `loop_z` loop, the combination of the P_i parameters and the FLOPs were measured using PAPI on an Intel Pentium III processor. There is a small number of different combinations of P_i parameters that can be feasible, since most of the parameters have similar physical meaning. For each of this parameter combinations, the arithmetic mean of all FLOPs associated was performed. Before an analysis of the results, because of the fact that some P_i parameters always have the same behavior, it is possible to reduce the number of necessary parameters for each different execution flow. Eventually, only four parameters ($Q_i, 0 \leq i \leq 3$) are necessary to characterize the main execution flow of the `rxn` routine. For each node of the 3D simulating mesh, the relationship between the combination of Q_i parameters and the FLOPs associated with each node (F_k) can be summarized in the simple algorithm shown in Fig. 4. Obviously, these numerical factors have a close dependency on the microprocessor architecture. However, the same procedure can be used to obtain these numerical factors for any other architecture.

With this information, and knowing the number of executions of the `rnx` routine for each node of the 3D simulating mesh, the number of FLOPs associated with each node of the simulating mesh can be estimated in a straightforward way. As the Q_i parameters can be calculated before starting the simulation, a map of

```
if (Q0 == 0)    Fk = 7715
else
   if (Q1 == 0 && Q2 == 0)    Fk = 10337
   else
      if (Q1 == 0 && Q2 == 1)    Fk = 2500 · Q3 + 13000
      if (Q1 == 1 && Q2 == 0)    Fk = 2440 · Q3 + 12400
      if (Q1 == 1 && Q2 == 1)    Fk = 4150 · Q3 + 15700
```

Fig. 4. Relationship between the Q_i parameters and the FLOPs associated with each node of the simulating mesh

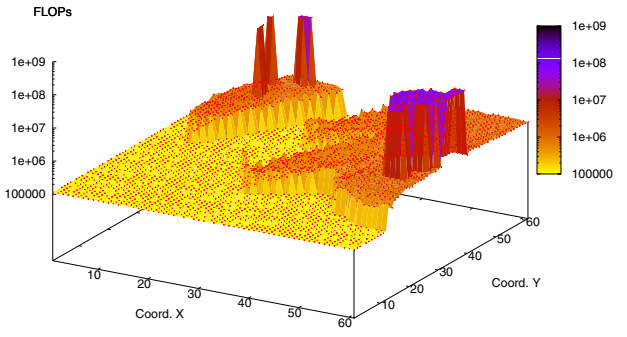


Fig. 5. Predicted FLOPs throughout the horizontal spatial dimensions

the predicted FLOPs throughout the 3D simulating space can be obtained just before starting the simulation. Fig. 5 shows an example where, for simplicity, the vertical spatial dimension was projected into the horizontal spatial dimensions. That is:

$$L(x_1, x_2) = mend \times \sum_{k=1}^{Z} F_k \tag{4}$$

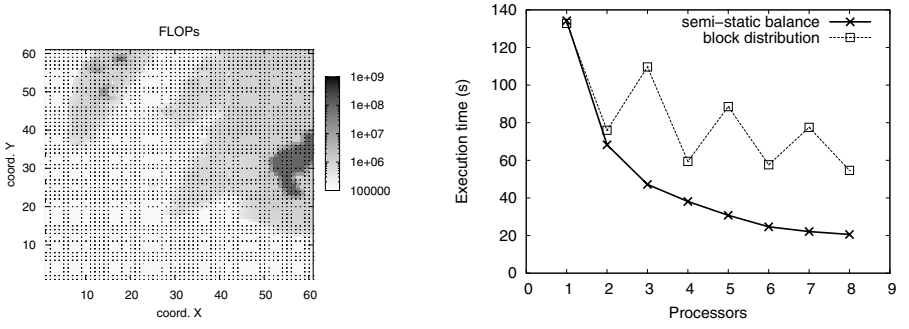
where $L(x_1, x_2)$ is the predicted FLOPs associated with a (x_1, x_2) 2D coordinate in the mesh, Z is the size of the vertical spatial dimension and *mend* is the number of times the `rxn` routine is executed in the (x_1, x_2) coordinate.

4.2 Load Balancing Strategy for STEM-II

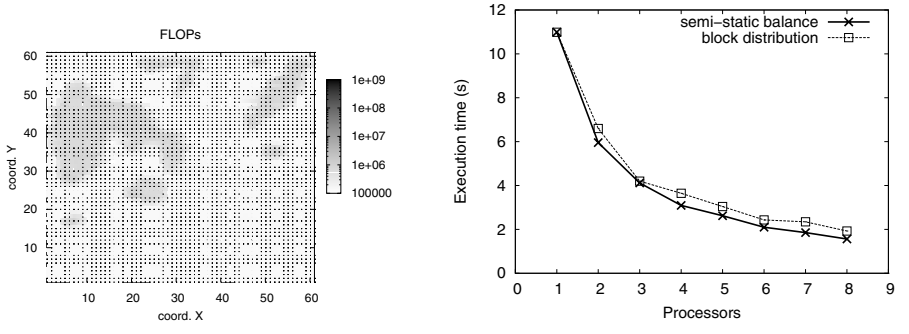
The proposed load balancing strategy is applied assuming that (4) supplies the computational load associated with a meteorological situation of each (x_1, x_2) coordinate of the 2D simulating mesh and considering that all the processors have the same computational power. As the meteorology changes during the simulation, this is a semi-static strategy.

For each meteorological situation the load distribution among the processor mesh is performed in two steps, as described in Sec. 3. In the first one, the columns of the simulating 2D mesh are distributed among the columns of processors. In the second step, each of the partitions performed in the first step is distributed among all the processors associated with the corresponding column of the processor mesh.

Fig. 6 shows the execution time in two different meteorological situations in a beowulf cluster. Only a load balance in one dimension was performed to simplify the example. Note that meteorological situations in which there are zones with high water concentration, like in Fig. 6(a), consume much more time than dry situations, like in Fig. 6(b). The achieved improvement with this semi-static strategy is critical in situations where there is a high concentration of water not homogeneously distributed among the simulating area (Fig. 6(a)). In other meteorological situations, in which the computational load is more homogeneous,



(a) Meteorological situation 1



(b) Meteorological situation 2

Fig. 6. Execution time of semi-static balance and block distribution, in two different meteorological situations, represented by its computational map throughout the horizontal dimensions of the simulating mesh

there is an improvement but not such a dramatic one (Fig. 6(b)). A homogeneous distribution of the water concentration is the only meteorological situation where the proposed strategy would not achieve any improvement. Therefore, optimization of the load balance for meteorological situations with a high inhomogeneous water concentration is critical to obtain useful predictions of this application.

5 Conclusions

This work presents a methodology to improve the load balancing in compute-intensive applications by means of computational models. This methodology is divided in two steps. A computational model of the application is performed in the first one. This model provides a relationship between the computational effort and the input data values throughout the simulating space and it is application dependent. This model can be used in schedulers and prediction tools in order to take decisions in heterogeneous environments. A load balancing strategy based on the computational model is performed in the second step. We propose

a heuristic that performs the distribution of the simulating space dimension by dimension. This methodology was applied to STEM-II application thereby obtaining an important improvement in execution time in comparison to the static block distribution, specially in situations in which the computational load is not homogeneously distributed throughout the simulating space.

Acknowledgments

This work was supported in part by the European Union through the IST-2001-32243 project "CrossGrid", and by the Ministry of Science and Technology of Spain through the TIN2004-07797-C02 project. The authors thank the supercomputer facilities provided by CESGA.

References

1. Ian Foster, Carl Kesselman. The Grid2: Blueprint for a New Computing Infrastructure. Elsevier, Inc. 2004
2. A. García Loureiro, K. Kalna, J. M. López González, A. Asenov. Three-Dimensional Simulation of InGaAs/AlGaAs pHEMT. Conferencia de Dispositivos Electrónicos. Barcelona, Spain, 2003
3. G. R. Charmichael, L. K. Peters, R. D. Saylor. The STEM-II Regional Scale Acid Deposition and Photochemical Oxidant Model-I. An Overview of Model Development an Applications. Atmospheric Environment **25** (1991) 2077–2090
4. CrossGrid project. http://www.crossgrid.org
5. M. Boullón, J. C. Cabaleiro, R. Doallo, P. González, D. R. Martínez, M. J. Martín, J. C. Mouriño, T. F. Pena and F. F. Rivera. A Performance Prediction Tool for Grid Enable Applications. To be published in L.N.C.S.
6. S. Browne, J. Dongarra, N. Garner, G. Ho, P. Mucci. A Portable Programming Interface for Performance Evaluation on Modern Processors. The International Journal of High Performance Computing Applications **14** (2000) 189–204
7. J. C. Mouriño, M. J. Martín, P. González, M. Boullón, J. C. Cabaleiro, T. F. Pena, F. F. Rivera and R. Doallo. A Grid-Enable Air Quality Simulation. First European Across Grids Conference. Santiago de Compostela, Spain, February 13-14, 2003
8. M. J. Martín, J. C. Mouriño, F. F. Rivera, R. Doallo and J. D. Bruguera. High Performance Air Pollution Modeling for a Power Plant Environment. Parallel Computing, 2003

Providing Autonomic Features to a Data Grid

María S. Pérez, Alberto Sánchez, Ramiro Aparicio,
Pilar Herrero, and Manuel Salvadores

Facultad de Informática, Universidad Politécnica de Madrid, Madrid, Spain

Abstract. Autonomic and Grid computing are complementary, in the sense that complex grid environments can take advantage of the features provided by autonomic computing and on the other hand, autonomic processes can be properly deployed by using grid technology. Besides, one of the most active fields in grid computing is the area of data grids, focusing on the data access. Our paper proposes a grid framework which provides autonomic characteristics in order to enhance the performance of the data access, predicting the future behaviour of the corresponding I/O system. The use of the autonomic system is transparent to the user. This paper also presents a study case of such system.

Keywords: Autonomic computing, Grid computing, Data grids, future behaviour prediction.

1 Introduction

Most of the progress made in Computer Science have arisen as result of the existence of some specific crisis, which implies a revolution in the corresponding research area. For instance, the known software crisis [2], whose notion emerged at the end of the 1960s, and which is characterized by an inability of software developers to deliver good quality software products according to the scheduled time and budget, caused the beginning of the software engineering [15].

In the same way, the I/O crisis and, more recently, the software complexity crisis, have been used for naming situations in which the current technology did not solve the problems originated by such crisis.

The I/O crisis is given by the difference between the computation and the I/O capacity, that leads to become the I/O system in a "bottleneck" in the nowadays systems [8]. On the other hand, a new software complexity crisis has been detected in current software systems [5]. These systems are so complex than their administration is becoming increasingly unmanageable. Both problems or crisis have not been properly solved, although there exist some initiatives for their resolution. In the first area, many different proposals have been provided. Parallel I/O systems field is one of the most active in this sense. In the latter one, the growing proliferation of the autonomic computing can allow software developers and administrator to make easier the management and administration of complex software systems.

Autonomic computing [12] [4] is used to describe the set of technologies that enable applications to become more self-managing. Self-management involves self-configuring, self-healing, self-optimising, and self-protecting capabilities. The word *autonomic* has been borrowed from physiology; as a human body knows when it needs

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 337–346, 2005.
© Springer-Verlag Berlin Heidelberg 2005

to breathe, software is being developed to enable a computer system to know when it needs to reconfigure itself.

Autonomic computing deployment is one of the most promising areas in computer science. If the environment in which this discipline is used is a grid [3], the advantages would be even higher, due to the complexity of these environments.

Our work intends to combine solutions from parallel I/O systems and autonomic computing in order to optimize the performance of the I/O phase (which is critical) in data grids [1]. Autonomic computing provides self-management, which in this case corresponds to the self-configuration and self-optimising capabilities. With this aim, we propose MAPFS-Grid [10], whose autonomic system takes decisions about the data location, based on monitored data. As our goal is to increase the system performance not in a concrete time point, but in future actions and during a time period, decisions are made according to a statistic prediction algorithm [14].

The outline of this paper is as follows. Section 2 defines the foundations and characteristics of autonomic computing. Section 3 describes our proposal, an autonomic architecture for a data grid, which provides autonomic features to a I/O system. Section 4 shows a study case of the autonomic part of our proposal. Finally, Section 5 explains the main conclusions and outlines the ongoing and future work.

2 Autonomic Computing

The increasing complexity of current software infrastructures can slow down the progress of the technology development. As transitions from the *information age* to the *knowledge era*, it seems clear that the need for data processing capabilities will continue growing in a exponential way. The huge amount of data we have to deal with everyday is impossible to be managed with current management systems. Nowadays applications require both a huge amount of computing capability and tools which make easier their configuration and deployment.

These two aspects are both sides of the same coin, which involves two innovative fields, namely grid computing [3] and autonomic computing [7]. Indeed, the intended goals of both areas can be seen as instances of a more general goal, that is, the usability of computing elements in a virtual environment. The main metaphor of this feature is the use of the telephony or electricity. These scenarios provide automated and standardized ways of using services, whose complexity is hidden for end users.

In [5] is emphasized the urgency of "...design and build computing systems capable of running themselves, adjusting to varying circumstances, and preparing their resources to handle most efficiently the workloads we put upon them. These autonomic system must anticipate needs and allow users to concentrate on what they want to accomplish rather than figuring how to rig the computing systems to get them there ..."

Autonomic computing tries to emulate the autonomic nervous system of a human body. The autonomic nervous system is the responsible for performing body tasks such as control the heart beating, check the blood's sugar and oxygen levels, monitor body's temperature, manage the food digestion and so on. And all these task are made in a unconscious fashion. In fact, this is the key feature that autonomic computing aims at achieving: the self-configuration is made without any conscious recognition by the user or developer.

In order to focus on this paradigm, it is important to understand the nature of autonomic computing. In [5], IBM, one of the most active supporters of autonomic computing (they were the first in coining this term), defines the following eight key elements of this discipline:

1. "To be autonomic, a computing system needs to 'know itself' - and comprise components that also possess a system identity". An autonomic computing system is aware of all the components and their status.
2. "An autonomic computing system must configure and reconfigure itself under varying and unpredictable conditions". The environment in which an autonomic computing system works is dynamic, and according to these dynamic conditions, the autonomic computing system must be able to reconfigure itself. Although the conditions are unpredictable, it is possible and desirable to use a system which can predict, in some sense the future behaviour. In this way, the configuration makes feasible the performance enhancement.
3. "An autonomic computing system never settles for the status quo - it always looks for ways to optimize its workings". An autonomic computing system monitors the overall status of the system and decides, according to a optimization plan, the parameters to be changed.
4. "An autonomic computing system must perform something akin to healing - it must be able to recover from routine and extraordinary events that might cause some of its parts to malfunction". An important feature of an autonomic computing is its ability for self-healing: a system must be able to identify the problems causes and solve them.
5. "A virtual world is no less dangerous that the physical one, so an autonomic computing system must be an expert in self-protection" An autonomic computing system must prevent itself from attacks, detecting them and alerting system administrator in case of danger.
6. "An autonomic computing system knows its environment and the context surroundings its activity, and acts accordingly". An autonomic computing system must be able to discover resources and obtain information about them. Furthermore, according to the information of its neighbours, the system takes decision.
7. "An autonomic computing system cannot exist in a hermetic environment". An autonomic computing system interact in an open and heterogeneous environment with other elements by means of open standards. This feature is especially compatible with the grid phylosophy.
8. "Perhaps most critical for the user, an autonomic computing system will anticipate the optimized resources needed while keeping its complexity hidden". An autonomic computing system must be able to act in advance in a optimized fashion in order to increase the performance of the system. This ability must be performed in a transparent way.

To achieve all these features, four generic principles are embedded into autonomic computing strategy, namely [7]:

- self-configuration, that is, the ability for configuring itself according to high level policies;
- self-optimisation, that is, the capacity of seeking ways of enhancing the performance;

- self-healing, that is, the feature which allows the system to detect, diagnose and repair hardware and software problems;
- self-protection, that is, the ability for preventing the system against possible attacks.

3 MAPFS-Grid: An Autonomic Data Grid Architecture

The difficulty and size of current problems involve a challenge for researchers, which need to use complex solutions and architectures to their domain-based problems. Grid computing has become a key piece for the development and deployment of these infrastructures.

Often, the higher complexity is due to the huge amount of data involved in such processes. In these scenarios, the I/O access stage limits the overall performance of the system. Furthermore, the optimum configuration of these environments is not usually straightforward.

MAPFS-Grid is a grid-based framework, whose main goal is to enhance the performance of data grid applications. Moreover, MAPFS-Grid is composed of an autonomic system, which is in charge of providing autonomic capabilities to the system.

Our optimization depends on the kind of I/O operation. In the case of write operations, MAPFS-Grid uses a prediction algorithm, based on logs and historic data, together with a decision policy to find out the "best" target cluster[1]. This is due to the fact that we use replicated data, in order to provide both fault tolerance and performance enhancement, and thus, although the written data are present in the system, we can choose an alternative location. A coherence protocol is used for updating all the data copies.

On the other hand, in the case of read operations, MAPFS-Grid optimizes the data access, depending on the current performance of all the locations where data are stored.

Before analysing every component of MAPFS-Grid, it is important to emphasize several aspects of such framework:

- MAPFS-Grid resources are clusters of workstations/servers or individual nodes. In general, any computation element with disk capacity can be considered a resource in our environment. This feature makes flexible the definition of resources in MAPFS-Grid.
- MAPFS-Grid is based on a multiagent parallel file system, named MAPFS [9], whose main contribution is the conceptual use of agents to provide applications with new properties, with the aim of increasing their adaptation to dynamic and complex environments. MAPFS offers features such as data acquisition, caching, prefetching and use of hints. MAPFS is intended to use in a cluster of workstations.
- In MAPFS, data is striped between the nodes of the clusters, in order to take advantage of the inherent paralellism of such layout.
- Associated to every resource (cluster or computing element) there is a grid service, with two main portTypes[2]: access and monitoring. The first one is the portType used for the main operations of the file system, and is explained in detail in [11]. The second one will be explained later.

[1] This is the best target cluster according to the used heuristics.

[2] The grid and web services communities are converging. Thus, the term portTypes is used in both disciplines.

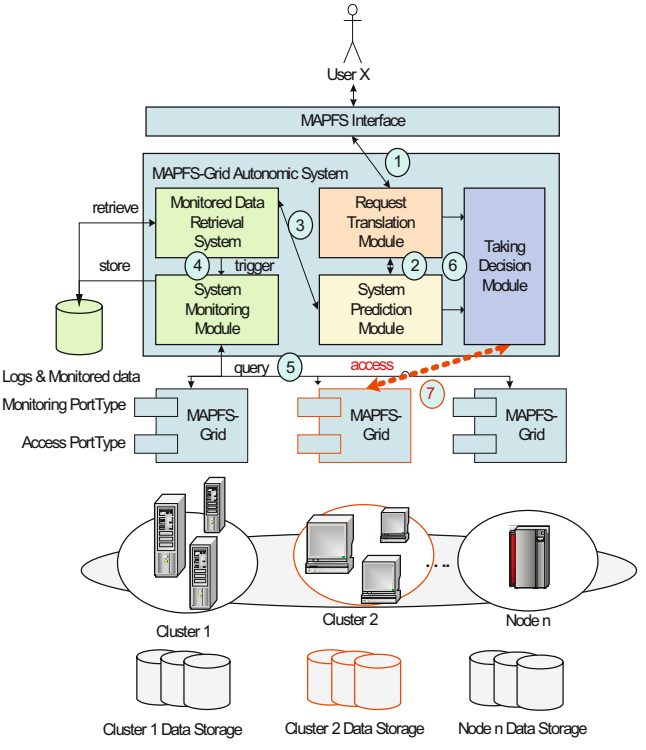


Fig. 1. MAPFS-Grid Architecture

Figure 1 shows the MAPFS-Grid architecture, focusing on the internal design of the MAPFS-Grid Autonomic System. Thus, the main components of MAPFS-Grid are:

- MAPFS Interface: MAPFS-Grid shares the interface with MAPFS. MAPFS provides a POSIX-like interface with advanced operations. Unlike MAPFS, which is used in a cluster, MAPFS-Grid is used in a grid composed of cluster and/or individual nodes.
- MAPFS-Grid Autonomic System: This is the autonomic part of MAPFS-Grid. It is composed of five modules, whose main responsability is providing autonomic features to the I/O management. Mainly, this component is in charge of taking decisions about the target cluster for an I/O operation, and aspects related to the data layout and management. The five modules of the autonomic system are:
  1. Request Translation Module: This module translates a MAPFS I/O operation in a request to the system prediction module (step 1 and 2). Basically, this module analyses the kind of operation and accordingly, it requests System Prediction Module the optimum storage element. In some operations, such as read routines, the system prediction module only provides the "best" storage element in which data are stored, without performing any prediction task. In create and write operations, the prediction task must be invoked.

2. System Prediction Module: In order to calculate the optimum storage element, the System Prediction Module queries monitored data (step 3) and uses a prediction method based on Markov chains, which is described in [14]. Basically, we use a probabilistic model based on past behaviour, which takes into account two types of parameters, measured at storage element level (cluster or individual node), that is: (i) Basic parameters, such as capacity of the hard disk of each server, network load (busy rate of the network), workload or disk bandwidth; (ii) Advance parameters, which are configuration parameters which affect to the performance of the autonomic system. One of the most significant advance parameters is the time window (T), that is, the time period in which the system monitors its performance.
3. Monitored Data Retrieval System: This module retrieves the data queried by the System Prediction Module from the logs and monitored data storage. In case this information is not available (this happens every T seconds), this module triggers the execution of the System Monitoring Module (step 4).
4. System Monitoring Module: Basic parameters are monitored by this module with the aim of improving the decisions taken by the autonomic system. This is made querying the Monitoring portType of every grid service (step 5).
5. Taking Decision Module: According to the predicted system state (step 6), this module decides the target cluster and the action to be done (step 7).

– MAPFS-Grid Service, with two portTypes, the access and the monitoring portype. As mentioned previously, the access portType is used for performing the file systems operations. On the other hand, the monitoring portType is used in order to obtain measures related to the storage element. A basic monitoring portType, whose functionality is querying performance parameters, is shown in Figure 2. The access portType is described in [11].

The Monitoring portType implementation uses MonALISA [6] and Ganglia [13]. MonALISA is a distributed monitoring system based on JINI/Java and WSDL/SOAP, whose main goal is providing information about large-size distributed systems. MonALISA provides a web service interface and allows existing monitorization tools to be integrated. We have used Ganglia as scalable monitorization tool for high performance distributed systems, such as clusters or grids.

```
<wsdl:portType name="MonitoringPortType">
   <wsdl:operation name="query">
      <wsdl:input message="tns:queryInputMessage"/>
      <wsdl:output message="tns:queryOutputMessage"/>
   </wsdl:operation>
</wsdl:portType>
```

Fig. 2. A very basic Monitoring PortType

4 Study Case

This section shows some results obtained by our autonomic system, which allow us to extract some interesting conclusions that assert our previous proposals. Through this analysis, our aim is predicting the future behaviour of our system in order to take the best decision that enhances the system performance.

Although MAPFS-Grid autonomic system is able to measure several parameters from several clusters which belong to a grid environment, for the sake of simplicity, we have decided to use only one parameter, the CPU load and two nodes Intel Xeon 2.40GHz with 1GB of RAM memory interconnected by means of a 2 Gigabit network. The CPU load is measured supporting a normal workload. We have considered that in our data grid, the CPU load is a crucial parameter, since the server runs the process request. These results can be extrapolated to a more complex environment, with different parameters (disk bandwidht, network load, etc.), depending on the specific requirements of the system.



Fig. 3. CPU load in node 1 during a time interval of 4 hours

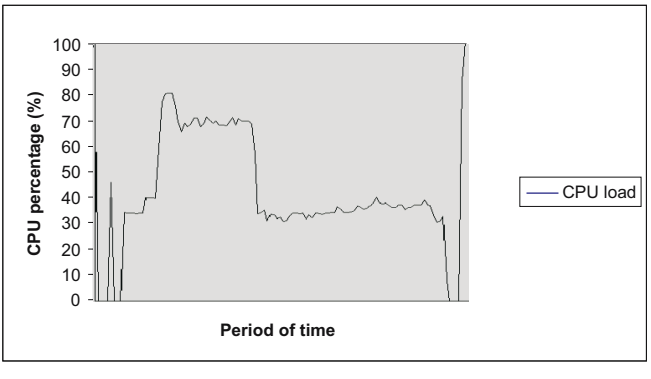


Fig. 4. CPU load in node 2 during a time interval of 4 hours

Figures 3 and 4 show the nodes workload, measured by means of the system monitoring module. Apparently, we cannot decide at first sight which is the best node where it would be advisable to write in order to improve the performance of the next I/O requests.

Table 1. Initial matrix of probabilities of transition between states for node 1

	$0-25\%$	$25-50\%$	$50-75\%$	$75-100\%$
$0-25\%$	42.0%	1.0%	0.0%	0.0%
$25-50\%$	1.0%	7.0%	1.0%	0.0%
$50-75\%$	0.0%	0.0%	0.0%	2.0%
$75-100\%$	0.0%	1.0%	1.0%	55.0%

Our system prediction module uses a Markovian approach, as explained in [14]. With the aim of simplifying the resolution of the the Markovian problem, we have defined four different states according to the CPU load percentage ($0-25\%$, $25-50\%$, $50-75\%$ and $75-100\%$). The system will be in this corresponding state when the computer workload is between the two suitable measures. The initial matrix 1 and 2 of probabilities of transition between states are obtained by means of collected data, shown in Figures 3 and 4. The time window has been setup as 2 minutes. Thus, we can obtain the probability of changing among states or maintaining the same state every 2 minutes.

We have solved this problem by means of the proposed Markovian approach, obtaining the corresponding solution vectors 3 and 4. By comparing both vectors, we

Table 2. Initial matrix of probabilities of transition between states for node 2

	$0-25\%$	$25-50\%$	$50-75\%$	$75-100\%$
$0-25\%$	4.0%	1.0%	0.0%	0.0%
$25-50\%$	0.0%	71.0%	2.0%	0.0%
$50-75\%$	0.0%	0.0%	24.0%	1.0%
$75-100\%$	1.0%	1.0%	0.0%	5.0%

Table 3. Solution vector for node 1

$0-25\%$	$25-50\%$	$50-75\%$	$75-100\%$
38.76%	8.1%	1.8%	51.3%

Table 4. Solution vector for node 2

$0-25\%$	$25-50\%$	$50-75\%$	$75-100\%$
4.31%	62.90%	22.40%	10.35%

can select the best resource that maximizes the expected remuneration in a further future. This remuneration depends of the used policy. In this sense, if we use a defensive attitude, we should select the second node to be written in order to improve the I/O requests in a further future because the node 1 has larger probability to stay in a worse state. However, if we use an aggressive policy, we could select the node 1, because the probability to stay in the best state $(0 - 25\%)$ is higher.

Nevertheless, since we are using heterogeneous environments, it is necessary to take into account that different computers can have different characteristics. In short, it would be advisable to define some rules to compare different nodes. For instance, in the case of measuring workloads, it could be interesting to multiply the CPU speed and the probability to stay in a concrete state. This value would represent the expected CPU speed in such state.

5 Conclusions and Future Work

This paper has shown MAPFS-Grid autonomic system, whose main goal is to provide autonomic features to data applications on grid environments. This system is composed of several modules: (i) Request Translation Module, which translates a MAPFS I/O operation in a request to the system prediction module; (ii) System Prediction Module, which calculates the optimum storage element; The basis of this module is explained in detail in [14], where a Markov-based prediction algorithm is described. (iii) Monitored Data Retrieval System, which retrieves the data queried by the System Prediction Module from the logs and monitored data storage; (iv) System Monitoring Module, which is in charge of monitoring information and stores on the corresponding database; and (v) Taking Decision Module, which decides the target cluster and the action to be done.

An analysis of the results obtained by our autonomic system is also presented at the end of the paper. For the sake of simplicity, these results are based on only one of the parameters that has influence about the data applications. As future work, we are planning to introduce both new parameters and rules for taking decisions. Furthermore, it would be desirable to define several policies, which allow us to take decisions, more or less agressive.

Acknowledgements

This research has been partially supported by Universidad Politécnica de Madrid under Project titled “MAPFS-Grid, a new Multiagent and Autonomic I/O Infrastructure for Grid Environments”.

References

1. A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke. The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets. *Journal Of Network And Computer Applications*, 23(3):187–200, 2000.
2. Edsger W. Dijkstra. The humble programmer. *Commun. ACM*, 15(10):859–866, 1972.

3. I. Foster and C. Kesselman, editors. *The Grid 2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 2004.
4. IBM and autonomic computing. An architectural blueprint for autonomic computing, April 2003.
5. IBM's Perspective on the state of information technology. *http://www-1.ibm.com/industries/government/doc/content/resource/thought/278606109.html*.
6. H.B. Newman, I.C. Legrand, P. Galvez, R. Voicu, and C. Cirstioiu. MonALISA: A distributed monitoring service architecture. In *Proceedings of CHEP, La Jolla, California*, March 2003.
7. Jeffrey O. Kephart, Davis M. Chess, and Thomas J. Watson. The Vision of Autonomic Computing. *IEEE Computer Society*, 2003.
8. D. A. Patterson, G. Gibson, and R. H. Katz. A case for Redundant Arrays of Inexpensive Disks (RAID). In *Proceedings of ACM SIGMOD*, pages 109–116, June 1988.
9. María S. Pérez, Jesús Carretero, Félix García, José M. Peña Sánchez, and Victor Robles. A flexible multiagent parallel file system for clusters. In Peter M. A. Sloot, David Abramson, Alexander V. Bogdanov, Jack Dongarra, Albert Y. Zomaya, and Yuri E. Gorbachev, editors, *International Conference on Computational Science*, volume 2660 of *Lecture Notes in Computer Science*, pages 248–256. Springer, 2003.
10. María S. Pérez, Jesús Carretero, Félix García, José M. Peña Sánchez, and Victor Robles. MAPFS-Grid: A flexible architecture for data-intensive grid applications. In F. Fernández Rivera, Marian Bubak, A. Gómez Tato, and Ramon Doallo, editors, *European Across Grids Conference*, volume 2970 of *Lecture Notes in Computer Science*, pages 111–118. Springer, 2003.
11. María S. Pérez, Alberto Sánchez, Pilar Herrero, and Víctor Robles. *Engineering the Grid*, chapter A new Approach for overcoming the I/O crisis in Grid environments. American Scientific Publisher, 2005.
12. IBM Research Autonomic Computing. *http://www.research.ibm.com/autonomic/*.
13. F. D. Sacerdoti, M. J. Katz, M. L. Massie, and D. E. Culler. MonALISA: A distributed monitoring service architecture. In *Proceedings of the IEEE Cluster 2003 Conference*, 2003.
14. Alberto Sánchez and María S. Pérez. A mathematical predictive model for an autonomic system to grid environments. In Osvaldo Gervasi et al., editor, *ICCSA (3)*, volume 3482 of *Lecture Notes in Computer Science*, pages 109–117. Springer, 2005.
15. Anthony Ira Wasserman. A top-down view of software engineering. *SIGSOFT Softw. Eng. Notes*, 1(1):8–14, 1976.

TCP Performance Enhancement Based on Virtual Receive Buffer with PID Control Mechanism*

Byungchul Park[1], Eui-Nam Huh[2], Hyunseung Choo[1], and Yoo-Jung Kim[3]

[1] School of Information and Communication Engineering, Sungkyunkwan University
roman07@hanmail.net, choo@ece.skku.ac.kr
[2] Division of Information and Communication Engineering, Seoul Women's University
huh@swu.ac.kr
[3] IT Infrastructure Division, National Computerization Agency
yjkim@nca.or.kr

Abstract. TCP is the only protocol widely available for reliable end-to-end congestion-controlled network communication, and thus it is the one used for almost all communications. Unfortunately, TCP is not designed with high-performance networking and computing. Thus the research for TCP to obtain good throughput in high-performance networking and computing is in progress all over the world actively. In this paper, we propose a new scheme which makes a TCP system achieve high throughput even with small buffer. The receive buffer almost empties due to the characteristic of original TCP but the amount of physical memory assigned for the buffer cannot be reduced because TCP flow control will downgrade TCP performance with the reduced buffer. However a TCP system applying our proposed scheme can reduce the size of physically assigned receive buffer without downgrading TCP performance. And then we use PID control mechanism as a tool to adjust the size of VRB properly. Lastly, we compare the throughput with two schemes, proposed scheme and original TCP scheme. As a result, the TCP using VRB obtains 46% higher throughput than the original one. And we also compare the amount of memory necessary for achieving the maximum throughput between two schemes. The result of second comparison shows that the proposed TCP spends 43% less memory than the tuned original TCP for same throughput.

1 Introduction

TCP is the only protocol widely deployed, for reliable end-to-end congestion-controlled network communication, and thus is most popular for almost all

* This work was supported in parts by Brain Korea 21 and the Ministry of Information and Communication in Republic of Korea. Corresponding author: Prof. H. Choo.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 347–356, 2005.
© Springer-Verlag Berlin Heidelberg 2005

communi-cations. Unfortunately, TCP is not designed for high-performance networking and computing - its original design decisions focus on long-term fairness first, and performance is considered secondary. Thus users must often perform tortuous manual optimizations simply to achieve acceptable behavior. The most important and often most difficult task to achieve optimization, is determining and setting appropriate buffer sizes. However, it is not easy to perform the task manually whenever the TCP is used, therefore research into methods used to obtain good throughput using TCP, is actively conducted throughout the world.

Dynamic Right Sizing (DRS), Automatic TCP Buffer Tuning (ATBT) and Linux auto-tuning are the most popular mechanisms deployed for enhancing TCP performance. These mechanisms resolve the problem of buffer size allocation, for improved performance without memory waste. Usually, a TCP system with a larger buffer obtains superior performance but consumes more memory, while a TCP system with a smaller buffer conserves memory, but results in poorer performance. Therefore study-ing mechanisms used to adjust TCP buffer sizes effectively is important for network researchers.

In this paper, a simple mechanism of adjusting TCP buffer size is not proposed, instead, a new scheme, which allows a TCP system to achieve high throughput even with a small buffer, is proposed. The scheme in this paper represents a more advanced scheme than existing schemes because the proposed scheme uses a smaller buffer for same throughput. The proposed scheme is especially effective for applications transferring bulk data. However, other applications not transferring bulk data do not suffer from extreme performance loss with the proposed scheme, and at worst, achieves the same performance as the original scheme.

and A PID control mechanism is used as a tool to properly adjust the size of the VRB. The VRB mechanism and PID mechanism readily combines since these two mechanisms work in a similar manner. The VRB is implemented using PID control on Linux kernel 2.4. Lastly, the throughput with two schemes is compared. As a result, the TCP using the VRB obtains 46% higher throughput than the original scheme. The amount of memory necessary for achieving maximum throughput between the two schemes is also compared. The results of the second comparison show that the proposed TCP consumes 43% less memory than the tuned original TCP while maintaining the same throughput.

The remainder of the paper is organized as follows. In Section 2, a review of the problems associated with TCP and PID are discussed. In Section 3, the VRB mechanism is defined and its operation is described with the PID controller and how it is organized. In Section 4, the performance of the VRB mechanism is evaluated with the PID controller. The final section presents the conclusions.

2 Related Works

2.1 TCP Flow Control

The TCP is a well known network protocol, used for reliable communication. There are two ways to achieve reliable communication. The first is to prevent

packet loss and the second is recover from every possible loss. The TCP chooses the former, and performs congestion control and flow control, limiting transmission rate and thus prevent packet losses. When the TCP detects a packet loss, it decides that the network is congested and performs congestion control. When the receiver does not have sufficient memory for receiving data from the sender, the TCP performs flow control.

The receiver lets the sender know the available receive buffer size through sending information regarding the receiver's capacity, piggy-backing on an ACK message. The available receive buffer size is computed as following. Available receive buffer size equals the value subtracting buffer size filled with data from total receive buffer size. Namely, the available receive buffer size means the size of unoccupied area of the receive buffer. This also means the receiver and the sender can adjust the transmission rate with information regarding the receivers capacity. Thus the sender does not over-whelm the receiver. This mechanism is described as TCP flow control. However, when communication is controlled with this mechanism, the communication may impact TCP throughput negatively, because TCP flow control basically limits the transmission rate. Therefore if the transmission rate is made higher by controlling the flow loosely, total network performance improves, even though the communication suffers some packet loss. The above sentence is proven by the experiments presented in Section 4 total network performance enhancement is focused on.

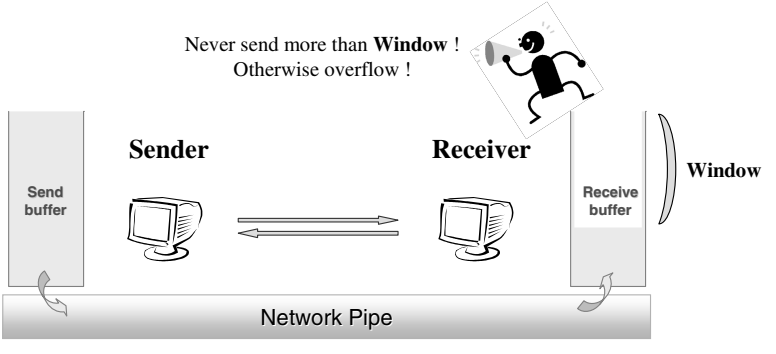


Fig. 1. Concept of flow control

2.2 PID Controller

PID can be described as a set of rules with which precise regulation of a closed-loop control system is obtained. This closed loop control represents a method in which real-time measurement of the process being controlled is constantly fed back to the controlling device, ensuring the desired value is realized. The mission of the control-ling device is to generate the measured value, usually known as the process variable, equal to the desired value, usually known as the set point. The most efficient method of accomplishing this task is with the use of the control algorithm, known as PID.

In its basic form, PID involves three mathematical control functions working together: Proportional-Integral-Derivative. The most important of these, proportional control, determines the magnitude of the difference between the set point and the process variable, known as error, applying appropriate proportional changes to the control variable to eliminate the error. Many control systems will, in fact, work reasonably well with only proportional control. Integral control examines the set point offset and the process variable over time, providing corrections when necessary. Derivative control monitors the rate of change of the process variable and consequently makes changes to the output variable to accommodate unusual changes.

Each of the three control functions is governed by a user-defined parameter. These parameters vary immensely from one control system to another, and need to be adjusted to optimize the control precision. The process of determining the values of these parameters is known as PID Tuning. In this paper, a PID controller is used to manage a virtual receive buffer, without performing PID tuning. Effective PID tuning is left as future work.

3 The Proposed Scheme

3.1 Motivation

In the original TCP flow control mechanism, a flow is controlled by transmitting the available receive buffer size from the receiver to sender. The goal of this flow control is to prevent overflowing the receive buffer, however, overflowing rarely occurs because the receive buffer is usually near empty because the CPU processing rate is higher than the data arriving rate. In other words, most receive buffer spaces are allocated wastefully, holding memory needlessly. That means the memory resource is not fully utilized. Therefore, attempt is made to improve the mechanism, utilizing memory effectively.

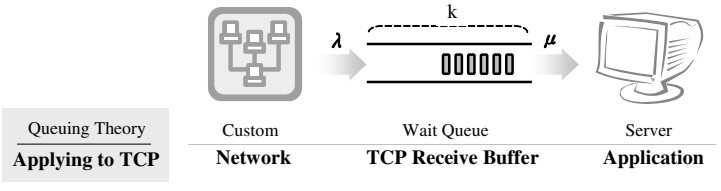


Fig. 2. Applying queuing theory to the TCP receive buffer

The TCP receive buffer corresponds to the M/M/1/K queuing model in queuing theory. The average length of the wait queue in this model can be calculated as the equation presented below. The parameters used in the equation are listed in Table 1, and the result, 0.0111, can easily be obtained with the table. This means the average length of the wait queue is 0.0111 of a byte, namely, the TCP receive buffer is filled with on average 0.0111 of a byte. The size of

Table 1. Parameters for queuing theory

	Definitions	Values
k	TCP receive buffer size	100000 byte
λ	Entrance rate of packet from network	10 Mbytes/sec
μ	Consuming rate of application	100 Mbytes/sec

the total receive buffer is ordinarily at least 10 Kbytes and thus the buffer filled with only 0.0111 of a byte can be considered a va-cant buffer. This implies the original TCP does not utilize its resource adequately. Considering good system efficiency, all system resources including the buffer should be fully consumed. In order to resolve this memory waste problem, a TCP virtual receive buffer mechanism is proposed, utilizing the TCP receive buffer more efficiently.

$$L_{ave} = \sum((i-1) \times (1 - \frac{\lambda}{\mu}) \times (\frac{\lambda}{\mu})^i). \quad (1)$$

3.2 VRB

The concept of virtual memory is applied in the general operating system to the TCP communication system. Since the TCP receive buffer is not fully utilized at once, as described in the previous subsection, the TCP does not need to assign physical memory for all buffers at a set time. Thus by assigning physical memory, as much as TCP requires at that point in time, the TCP can achieve high efficiency in terms of the receive buffer. Even though the TCP assigns as much memory as it needs, the receiver advertises a larger window than the assigned memory. This mechanism is defined as a Virtual Receiver Buffer (VRB) mechanism. Packet loss may occur when using the VRB scheme, since the TCP advertises a larger window than the actual one. However the total throughput of the proposed TCP scheme is much higher than the basic TCP scheme, because the gains from using the buffer efficiently outweigh the losses caused by packet loss.

Linux kernel 2.4.20-8 was chosen when implementing the VRB. The VRB mechanism should work on the receiver side the moment when a receiver advertises a receive window. The component performing the window advertisement in the kernel source must be located and modified for the VRB scheme to operate. Specifically, the tcp_transmit_skb function in the Linux Kernel is modified. The performance improvement using this modification is shown in Section 4.

3.3 PID on VRB

The VRB is a mechanism advertising window, if the receive window size may be very large even though the physically assigned buffer is small. As the result, the sender's transmission is not limited by flow control, and the sender can transfer

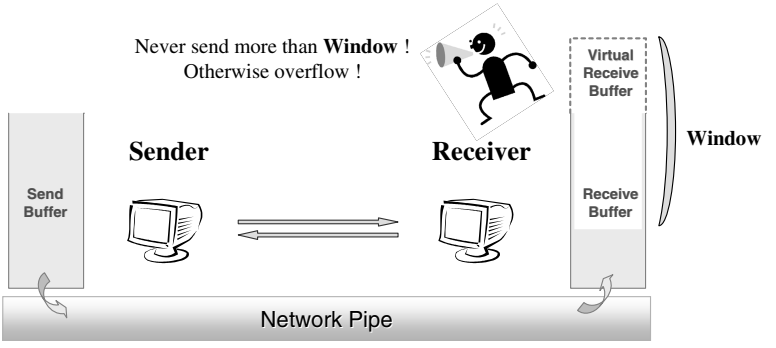


Fig. 3. Concept of VRB

data over high speed. However, a problem for deciding on the VRB size occurs. In order to resolve this problem, the PID control mechanism is applied to the VRB scheme. The total size of the physically assigned receive buffer corresponds to a set point within the PID controller, and the amount of received data occupied from network corresponds to the PID controller process variable.

If the TCP determines the virtual receive buffer size using the PID controller, then the size can be dynamically adjusted, according to the application or network state, since the PID controller naturally has the dynamic control ability. The PID controller decreases the virtual receive buffer size when network traffic is light and the receiver application does not consume received data frequently, because the available receive buffer mentioned in Subsection 2.1 becomes smaller from the network and the application. However, the PID controller increases the virtual receive buffer size when network traffic is heavy or the receiver application frequently consumes received data, because the available receive buffer becomes larger. Since the PID controller works dynamically in this manner, packets are hardly lost even though a larger value than the actual one is used for the window advertisement.

4 Performance Evaluation

4.1 Evaluation Environment

Two universities participate in the experiment for performance evaluation and com-municate with each other. The specifications of the hosts are listed in Table 2.

4.2 Experiments

Two experiments are conducted, to evaluate the performance of the VRB scheme. The first is for simple comparison of the throughputs and the second is for comparison of the throughputs and the changes ac-cording to the buffer size. All

Table 2. Specifications of the systems used for experiment

	X University in Seoul	Y University in Suwon
OS	Linux kernel 2.4.18	Linux kernel 2.4.20
CPU	Pentium III 651.468 MHz	Pentium III 803.440 MHz
RAM	190812 KB	384520 KB
LAN Card	100 Mbps Ethernet 1500 MTU	100 Mbps Ethernet 1500 MTU

tests are per-formed between the original and proposed TCP. The original TCP is the pure TCP in the Linux Kernel 2.4.20-8 applying no tuning technique, and the proposed TCP is the TCP improved through modifying the Linux Kernel 2.4.20-8 source to apply the VRB scheme controlled by the PID mechanism on the TCP. The main tool used in each experiment is IPERF, which is a program to measure maximum TCP bandwidth, al-lowing the tuning of various parameters. IPERF reports bandwidth, delay jitter, data-gram loss and so on. Useful information such as bandwidth will be used in presenting the result of the test, in the next subsection.

In the first experiment, the TCP receive buffer size is set to 30Kbytes where the VRB mechanism operates most efficiently. 30Kbytes are not large enough to obtain a good result in basic TCP, but the proposed scheme achieves good results since it is designed to work efficiently even with a small buffer. The PID parameters are set to values shown in the following Table 3. The throughput is then obtained by executing the IPERF program.

In the second experiment, variable sizes ranging from a small size of approximately 10Kbytes to large size of approximately 100Kbytes, were tested. Size adjustment can be achieved by setting IPERF parameters, indicating buffer size when the IPERF executes. The PID parameters are set the same as the first test and the result is ob-tained through the IPERF program.

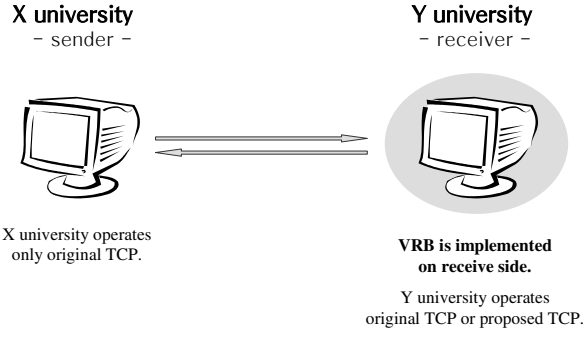


Fig. 4. Experiment environment

Table 3. PID parameters for experiment

	Definitions	Values
P	Coefficient of the proportional part	1.5
I	Coefficient of the integral part	0
D	Coefficient of the derivative part	0.2

4.3 Results Analyses

Fig. 5 demonstrates the result of the experiment comparing throughput between the basic TCP and the TCP using the VRB mechanism. The VRB mechanism improves TCP performance, in terms of throughput, by 46%. In the communication environment described in Subsection 4.1, the 30Kbytes buffer size is not sufficient in achieving good throughput for the basic TCP, but the size is sufficient in achieving high throughput for the TCP using the VRB scheme. Even though 30Kbytes is insufficient for a communication between two hosts which have a 100Mbps bottleneck and 3msec delay, the VRB mechanism allows the sender to transmit packets at high speed, with-out being limited by flow control since the VRB mechanism provides larger than ac-tual size, using a virtual buffer. However, several packets may be lost because the TCP loosely controls the flow. This occurs very infrequently, and can be controlled by TCP congestion control even if it occurs. In conclusion, the total throughput increases due to the high transmission rate of the sender regardless of packet loss.

Fig. 6 demonstrates not only throughput enhancement, but also throughput variation according to receive buffer size. The difference in the throughput between the two schemes is large, with a smaller buffer size of approximately 60Kbytes, because the VRB scheme attempts to exaggerate the small buffer size as if the size is larger than actual one. However, the difference in the performance of a larger buffer size to about 60Kbytes is almost zero because the receive buffer is already allocated sufficiently even when using the basic TCP. The buffer sizes required to achieve maximum throughput is different between the two

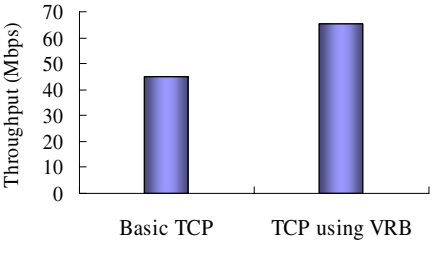


Fig. 5. Comparison of the performance between two schemes

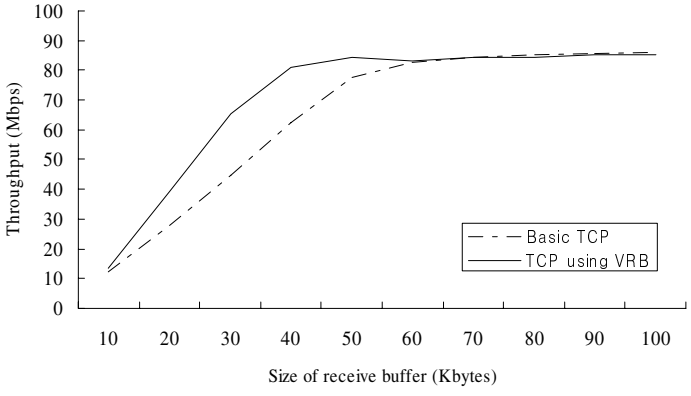


Fig. 6. Comparison of the throughput according to receive buffer size

schemes. While the original TCP requires 70Kbytes, the proposed TCP requires only 40Kbytes of memory. In other words, the proposed TCP can save about 30Kbytes of memory when the TCP attempts to obtain maximum throughput in this communication environment.

5 Conclusion

If memory size did not have any limitation, memory management is not required. However, physical memory must have a practical and physical boundary, thus managing memory efficiently is very important for system performance. In this paper, a mechanism managing memory efficiently by introducing the concept of a virtual receive buffer, and there-fore enhancing system performance is proposed. A PID control mechanism is also introduced to control the virtual buffer, and system performance is improved by setting PID parameters properly. However, improved performance may be achieved if PID parameters are deter-mined by PID tuning. As future work, an upgraded VRB mechanism will be developed using PID tuning.

References

1. Ann Chervenak, Ian Foster, C.Kesselman, C.Salisbury and S.Tuecke, "The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets," Journal of Network and Computer Application, vol.23, pp. 187-200, 2001.
2. Bill Allcock, Igor Mandrichenko, Timur Perelmutov, "GridFTP v2 Protocol Description," Germi National Accelerator Laboratory, 2004.
3. Deep Kakadia, "Understanding Tuning TCP," Sun BluePrints, March, 2004.
4. B. Tierney, "TCP Tuning Guide for Distributed Applications on WAN," In USENIX&SAGE Login, http://www-didc.lbl.gov/tcp-wan.html, February 2001.

5. J.Semke, J.Mahdavi and M.Mathis, "Automatic TCP Buffer Tuning," ACM SIGCOMM 1998, vol. 28, no.4, 1998.
6. W. Stevens, "RFC 2001: TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," January 1997.
7. V. Jacobson, R. Braden and D. Borman, "RFC 1323: TCP Extensions for High Perform-ance," May 1992.
8. Brian L. Tierney, Dan Gunter, Jason Lee, Martin Stoufer, "Enabling Network-Aware Appli-cations," IEEE-HPDC, 2001.
9. Mark K. Gardner, Wu-chun Feng, Mike Fisk, "Dynamic Right-Sizing in FTP (drs-FTP): Enhancing Grid Performance in User-Space," IEEE Symposium on High-Performance Dis-tributed Computing (HPDC-11/2002), Edinburgh, Scotland, July 2002. LA-UR 02-2799.
10. Eric Weigle and Wu-chun Feng, "Dynamic Right-Sizing:A Simulation Study," IEEE ICCCN, 2001.
11. Eric Weigle and Wu-chun Feng, "A Comparison of TCP Automatic Tuning Techniques for Distributed Computing," HPDC-11, 2002.
12. Matthew Mathis, Jeffrey Semke, Jamshid Mahdavi, Teunis Ott, "The Macroscopic Behavior of the Congestion Avoidance Algorithm," Computer Communications Review, volume 27, number 3, July 1997.
13. M. Mathis, J Heffner and R Reddy, "Web100: Extended TCP Instrumentation for Research, Education and Diagnosis," ACM Computer Communications Review, Vol 33, Num 3, July 2003.
14. Takahiro Mautsuo, Go Hasegawa and Masayuki Murata, "Scalable Automatic Buffer Tuning to Provide High Performance and Fair Service for TCP Connection," in proceedings of IEEE INET 2000, July 2000.
15. Pittsburgh Supercomputing Center, "Enabling High-Performance Data Transfers on Hosts," http://www.psc.edu/networking/perf tune.html.
16. Mike Fisk and Wu-chun Feng, "Dynamic Adjustment of TCP Window Sizes," http://public.lanl.gov/mfisk/fisk/web/papers/tcpwindow.pdf, July 2000.
17. Linus Torvalds and The Free Software Community, "The Linux Kernel," September 1991, http://www.kernel.org/.
18. Andras Veres and Miklos Boda, "The Chaotic Nature of TCP Congestion Control," in Proceedings of IEEE Infocom 2000, March 2000.
19. Mark Gates, Ajay Tirumala, Jon Dugan and Kevin Gibbs, Iperf, http://dast.nlanr.net/Projects/Iperf/, NLANR, 2003
20. Linux man page, http://www.die.net/doc/linux/man/man7/tcp.7.html

Computational Grid vs. Parallel Computer for Coarse-Grain Parallelization of Neural Networks Training

Volodymyr Turchenko

Research Institute of Intelligent Computer Systems, Ternopil State, Economic University, 3 Peremoga Square, 46004, Ternopil, Ukraine
vtu@tanet.edu.te.ua

Abstract. Development of a coarse-grain parallel algorithm of artificial neural networks training with dynamic mapping onto processors of parallel computer system is considered in this paper. Parallelization of this algorithm done on the computational grid operated under Globus middleware is compared with the results obtained on the parallel computer Origin 300. Experiments show better efficiency for computational grid instead of parallel computer with an efficiency/price criterion.

1 Introduction

Parallel data processing is one of the approaches that would make possible the execution of complicated algorithms with a large number of variables and iterations. A simple parallelism consists in dividing the sequential program among the available processors and run it in parallel in order to reduce the total execution time. However, this simple approach might not be effective due to additional overhead spent for the synchronization, communication and load imbalance between processors. As a result, several details related to the development of effective parallel programs remain urgent issues to deal with.

Artificial neural networks have excellent abilities to model difficult nonlinear systems. They represent a very good alternative to traditional methods for solving complex problems in many fields, including image processing, pattern recognition, robotics, etc [1]. However, most artificial neural network models require high computational load, especially in the training phase (up to days and weeks). This is, indeed, the main obstacle in front of an efficient use of neural networks in real-world applications, especially in real-time systems. Taking into account the parallel nature of artificial neural networks, many researchers have already focused their attention on the parallelization of different neural networks on several parallel systems [2], [3], [4], [5], [6]. However, there are still several bottlenecks for mapping neural networks onto the processors of the parallel machines [7], [8].

Among available today high-performance and distributed computing platforms, computational clusters and grids have gained tremendous popularity in computation science [9]. Grid computing enables the development of large scientific applications on an unprecedented scale [10]. Grid-aware applications, also called meta-applications or multidisciplinary applications use coupled computational recourses

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 357 – 366, 2005.
© Springer-Verlag Berlin Heidelberg 2005

that are not available at a single site. Grid technology lets scientists solve larger or new problems by pooling together resources that could not be coupled easily before [11], [12]. In order to enable grid computing, it has been recognized that a number of basic middleware services must be provided for such issues as authentication/security, information, resource access, data management, etc [13]. Many middleware packages [14], [15], [16], [17] were implemented by many teams among which Globus [18] is the most widely known. However, experiments with large configurations and real applications have shown that the latency of wide area networks is prohibitively high and that substantial bandwidth can hardly be achieved [12], [19]. Therefore an efficiency research of parallelization of different user-aware applications using grid technologies is an important and urgent research issue.

The main goal of this paper is to estimate an efficiency of coarse-grain parallel algorithm of neural networks training on computational grid with Globus middleware in comparison with parallelization results of the same algorithm achieved on parallel computer. As a case study we have used parallel algorithm of Integration Historical Data Neural Networks (IHDNNs) training with dynamic mapping of IHDNN modules onto processors of parallel computer [20]. Although we have already achieved the parallelization results of this task on parallel computer, the necessity to move to a grid system is proven by the fact, that it will be quite cheaper to develop computational grid based on existing computer infrastructure of the plant/site instead buying and setup a high performance computer with parallel architecture for a majority of practical applications in industry. Moreover wide usage of the Internet allows using computational grids remotely available worldwide providing minimization of time and financial expenses that is a crucial issue for an industry and business.

This paper is organized as follows. Section 2 introduces some details of the historical data integration method using neural networks in the context of previous works. Section 3 describes the development of coarse-grain parallel algorithm of IHDNNs training with dynamic mapping of neural networks modules onto processors of parallel machine using message-passing approach. Section 4 outlines a comparison of the parallelization results achieved on computational grid and parallel computer. Finally, Section 5 concludes this paper.

2 Sensor Drift Prediction Using Neural Networks and Choice of Parallelization Level

The sensor drift prediction is an important problem in intelligent data acquisition systems that are widely used in industry, environmental monitoring, in the space and in military applications [21]. One of the attractive features of such systems is their ability of providing some properties such as accuracy and self-adaptation to external exploitation conditions [22]. Sensors, as a first component of such systems, produce the largest error in the data acquisition process. The principal part of this error is due to the sensor drift during the sensor exploitation caused by the influence of external exploitation conditions. The main methods of drift reducing are calibration, testing and prediction. The calibration and testing, to be performed by the hardware, are very laborious for most of the modern sensors and, therefore, are rarely fulfilled [22]. The goal of prediction methods, which can be done by software, is reducing the number of

calibrations, i.e. increasing the inter-calibration interval. Knowing the prediction values of the sensor drift between calibrations, it is possible to fulfill the calibrations less frequently and to correct the current sensor readings.

The original method to form the IHDNN's training set [23] has been proposed in order to fulfill this task. The main advantage of this method consists in its ability to improve the accuracy of the sensor data acquisition and processing for several times (3-5 times) on increased duration of inter-calibration interval (6-12 times) using small number of input data as training set [24]. It has been proposed in [25] the use of an ensemble of three neural networks with different properties, which are connected sequentially, i.e. the output of the first neural network is considered as the input for the second one and so on. These neural networks are called Integrating Historical Data Neural Network (IHDNN), Approximating Neural Network (ANN) and Predicting Neural Network (PNN). More details about this method of Historical Data Integration can be founded in [24], [25], [26]. However, this method requires a considerable computational effort (approx. 40 minutes for one data acquisition channel on standard PC), a fact that makes its implementation on parallel computers an urgent task in order to ensure their use in real-life applications.

The use of real data of the sensor calibration is not expedient for IHDNNs investigation because real data do not fully describe the behavior of a sensor drift. Thus, mathematical models of sensor drift are usually developed for experimental researches. The results of industrial sensors calibrations in real environment were the basis of these mathematical models. The real data about the drift were supplemented by additional components that model non-stationarity and the non-uniformity of the drift, systematic and random errors of standard sensors, methodical errors, noises etc.

For the experiments we have used the model "with saturation" which corresponds to the drift of the thermo-resistor 30K5A1 at a working temperature of 150°C [25]. Average relative error of the historical data integration method did not exceed 15%. It allows receiving average and maximum relative errors of the sensor drift prediction not more than 9% and 31% respectively. These values correspond to improvement of the sensor data accuracy in 3 times on the inter-calibration interval increased in 12 times [24], [25], [26]. It is expediently to note, that the method of sensor drift prediction using neural networks outperforms the well-known mathematical prediction methods, in particular polynomials, curve-linear aligning and cubic spline [24].

It is necessary to have m copies of the IHDNN to predict m drift values of the new sensor. In [24] the author has shown that 5 values on the drift curve of the new sensor are enough to provide a training set for the ANN. The mathematical model of sensor drift is designed for 10 available curves of sensor drift gathered in the previous moments of time. This configuration of the input task (10 drift curves and 5 drift values on each curve) was used for experimental researches of accuracy of proposed historical data integration method in the past. Now this configuration of the input task is used only for the experimental researches of this parallelization method.

However in the practical applications, more drift values and more drift curves available provide an accuracy of sensor drift prediction considerably better. This fact confirms the necessity of redundant and high-performance computations using grid technology for this task. In a case of experimental research described in this paper we consider a scenario having 10 historical curves of the sensor drift for one sensor in one data acquisition channel. In practice such systems operates with tens and

hundreds data acquisition channels and the task of the neural network is to predict sensor drift for each sensor in real time scale. For example, it is used more than 200 sensors to control the technological parameters of the plane wing during its construction. Normally approximately 50^{th} historical data (50^{th} drift curves) could be considered for each sensor and data acquisition channel. Therefore the dimension of the real problem could be in 10^4 times greater than the case described in this paper.

We have used a modular neural network approach [27], which allows dividing a large problem into smaller tasks and fulfilling them by separate simple models of neural networks with the results reducing in the end of the whole algorithm. In our case we have 50 IHDNNs and each module of IHDNN uses its own data to form its training set [23]. The input data of each IHDNN do not depend on the input and/or the output of the other IHDNNs, each IHDNN is characterized by a high computational load. Therefore it is expedient to consider possible ways to parallelize this task.

Several approaches to parallelize neural networks have been proposed in the literature [5]: according to the architecture of the network, taking advantage of the matrix learning rule calculations [28], or parallelizing the presentation of examples [29]. In [30], three nested levels of parallelism in neural algorithms have been considered: connection parallelism (parallel execution on sets of weights), node parallelism (parallel execution of operations on sets of neurons), and example (modular) parallelism (parallel execution of examples on replicated networks). The first two levels are a fine-grain parallelism and the third level is a coarse-grain parallelism. Fine-grain parallel algorithms require a lot of low-level communications, for example to combine the results of the parallel calculation of weights of each neuron. Their use is more effective on processors arrays or network of transputers [3], [6]. Vice versa the coarse-grain algorithms are useful when big independent computation tasks should be processed and communications are rarely required. The use of high-performance computers with powerful parallel processors is recommended for the implementation of such parallel algorithms [30].

Therefore, taking into account the modularity of the input task, where each module presents separate IHDNN implementation, it is expedient to choose coarse-grain level of parallelism. As it is seen in the Introduction, there are still several bottlenecks for mapping neural networks onto the processors of the parallel machines [7], [8]. One of the possible approaches could be dynamic mapping scheme, where each processor that just finished executing of the current job on neural network training is scheduled immediately for another one neural network.

3 Parallel Training of IHDNNs with Dynamic Mapping

Parallel algorithm (Fig. 1) is developed by using a "centralized" planning approach with only one processor (*Master*, $cp = 0$) having the role of task planner and each of the other processors (called *Slaves*) will train the IHDNNs assigned by the *Master*. Once a *Slave* has finished its task, it asks the *Master* for a new one till no tasks are left. We note here that, besides the role of planner, the *Master* does not fulfill any further calculation. The communications between the *Master* and the *Slaves* are ensured by using the standard MPI sending/receiving functions *MPI_Send()/Resv()*.

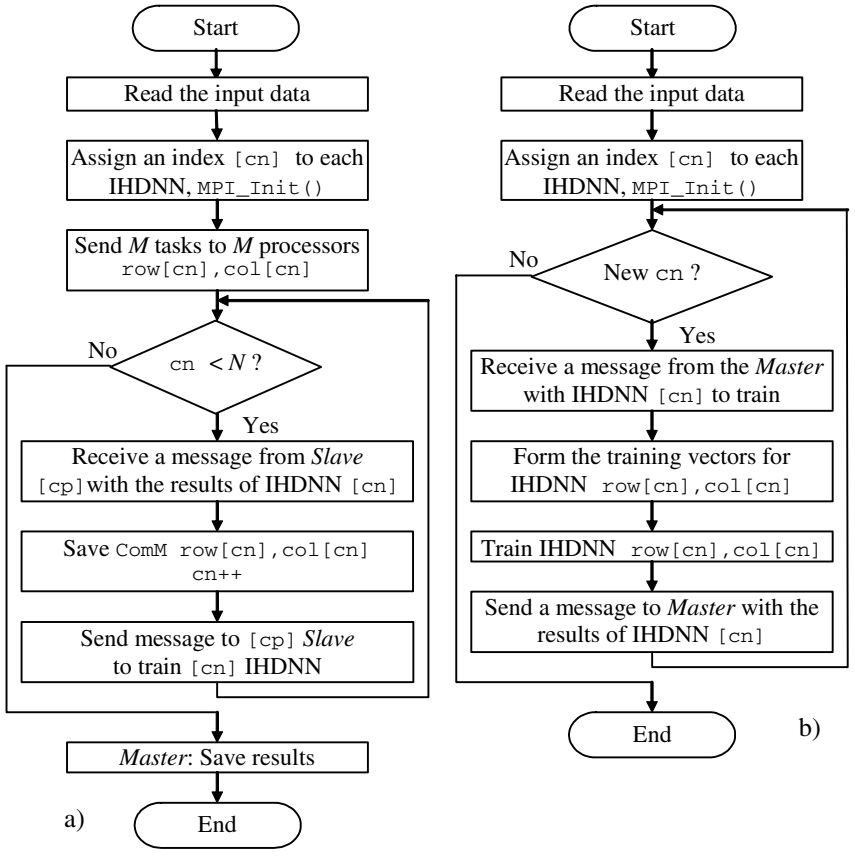


Fig. 1. Coarse-grain parallel algorithm of IHDNNs training with dynamical mapping: (a) *Master*'s procedure, (b) *Slave*'s procedure

The sequential part of the algorithm includes two operations: (i) reading the input data with the sensor drift and (ii) defining the sequential numbers of the IHDNNs based on the number of sensor drift curves and that of calibration points (Fig. 1a). The parallel part of the algorithm starts with the call of the *MPI_Init()* function (Fig. 1b). The index $< cn >$ denotes the sequential number of IHDNN and the $< cp >$ index refers to the processor ID. All the communications between the *Master* and the *Slave* include the index $< cn >$, which is useful for each process to pick up the data corresponding to the IHDNN to be trained. This scheme allows a considerable decrease of the length of the messages and consequently the latency time of the communications.

The *Master* (Fig. 1a) starts with assigning the first M IHDNNs to the M available processors and then continues the execution of the mapping procedure consisting in assigning dynamically the tasks to the *Slaves* as soon as they become idle. The *Master* receives also the results of the already trained IHDNNs by using the function *MPI_Recv()* with the *MPI_ANY_SOURCE* parameter and saves its content in the

appropriate cell of the output matrix $ComM$. The stopping condition of the algorithm is to check that all the IHDNNs have been already mapped.

Each *Slave* (Fig. 1b) checks the availability of a new message from the *Master* by using *MPI_Probe()*. Among all the received messages, each *Slave* should consider only those having an index $< cp >$ corresponding to its proper ID. On the basis of the IHDNN index $< cn >$ to be trained each *Slave* performs the following operations:

- Form the training set of the IHDNN based on the values $row[cn]$ and $col[cn]$ corresponding to the number of sensor drift curve and the sensor drift values respectively. The algorithm of training set forming is described in [23];
- Train the IHDNN as multi-layer perceptron using level-by-level back propagation training algorithm with adaptive learning rate;
- Run the historical data integration procedure and send the results, together with the IHDNN index $< cn >$, to the *Master*.

On the basis of the above description it is clear that our parallel algorithm with dynamic mapping does not use any synchronization point. Therefore, the only source of efficiency loss in our implementation can be derived from the overhead caused by the message passing communication. In the following section we are providing a comparison of performance assessment of the developed algorithm using both parallel computer and the computational grid.

4 Experimental Researches

The time of parallel routine running in comparison with the time of sequential routine running for training of 50^{th} IHDNN modules are measured during experimental researches. Several scenarios were investigated at training of each IHDNN module for sum-squared error (SSE) of neural network training changing from 10^{-3} to 10^{-7}. Each IHDNN (multi-layer perceptron) has 9 input neurons, 7 neurons in the hidden layer with a logistic activation function and one output neuron with a linear activation function. We have used level-by-level back-propagation training algorithm with adaptive learning rate modification on each training step. More details about IHDNN and its training algorithm are described in [23-26].

The time of parallel executing of IHDNNs is measured on 2, 4 and 8 processors of the parallel computer and the computational grid. The parallel algorithm has been developed by using C as programming language, MPI v.1.2 [31] as message passing library for the parallel computer, MPICH-G2 v.1.2.6 [17] as message passing library for the computational grid and MPE v.1.9.2 as performance visualization package.

The parallel computer Origin 300, installed in the Center of Excellence of High Performance Computing, University Calabria (Italy) has been used in this experimental research. Origin 300 has identical blocks Origin300_1 and Origin300_2. Each block consists of four 64-bit processors MIPS R14000 with a clock rate of 500 MHz and 4 Gb of local RAM. Each processor has a primary data and instruction cache of 32 Kb and the second level cache of 2 Mb. There are 4 RAM channels access in each

block. These two blocks are connected via high-speed NUMAlink interface. Origin 300 has a UNIX-based operating system IRIX 64 v.6.5.

For efficiency comparison of the developed parallel algorithm we have used the computational grid with Globus middleware [18]. The computational grid, also installed in the Center of Excellence of High Performance Computing, University Calabria (Italy), consists from 4 dual-processor personal computers Compaq ML350T01 with two Pentium III 933 MHZ processors, 128 Mb PC133 MHz RAM, integrated L2 cash 256 Kb, system bus clock rate 133 MHz, 9.1 Gb SCSI HDD, Fast Ethernet 100 Mbit/s network connection to 24-port 3Com Switch 100Mbit/s. Operation system of each computer is RedHat Linux 9 with Globus toolkit v.3.2.1.

The execution times of parallel IHDNNs training on the parallel computer Origin 300 and the computational grid are showed in Table 1 and Table 2 respectively. The speedup and efficiency of parallelization are showed in Fig. 2 for Origin 300 and in Fig. 3 for the computational grid.

It is necessary to note, that a computational complexity of each scenario of 50^{th} IHDNNs parallelization is the same on both computation platforms. Because each IHDNN module has the same input data and it trains to the same SSE value in both cases. It allows us to estimate a computation power of the both systems. For example, analyzing the Tables 1 and 2 we can see that a computational power of the computational grid at least in 4 times bigger in average in comparison with the parallel computer with considered hardware configurations of both systems.

Table 1. Execution time in seconds for Origin 300

CPUs	1	2	4	8
SSE=10^{-3}	23.41	12.63	6.62	3.58
SSE=10^{-4}	173.14	94.54	54.72	28.92
SSE=10^{-5}	637.70	429.22	321.94	248.69
SSE=10^{-7}	1114.53	668.15	452.31	326.39

Table 2. Execution time in seconds for the computational grid

CPUs	1	2	4	8
SSE=10^{-3}	9.48	4.78	2.45	1.29
SSE=10^{-4}	40.36	20.35	10.21	5.60
SSE=10^{-5}	169.17	85.21	64.58	55.16
SSE=10^{-7}	313.30	157.45	98.22	74.33

An efficiency comparison of the parallel algorithm on both computer systems is showed in Table 3. As it is seen, the average difference among both efficiencies is very small - 0.0067%. It allows making a conclusion that both parallel systems have practically the same efficiency for parallelization of coarse-grain algorithm of neural network training.

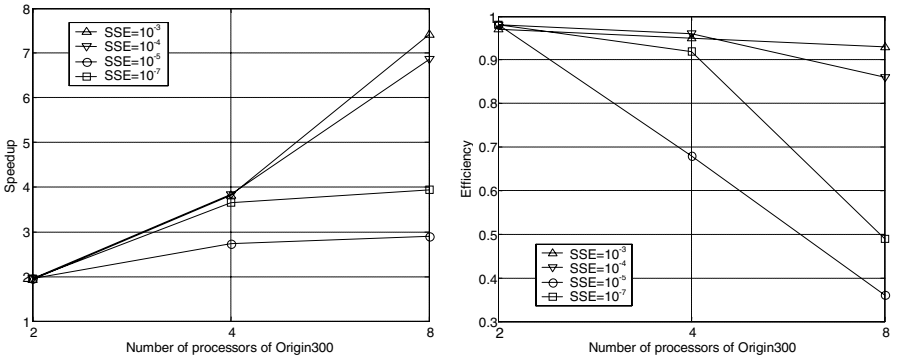


Fig. 2. Speedup and efficiency of parallel algorithm execution on parallel computer Origin 300

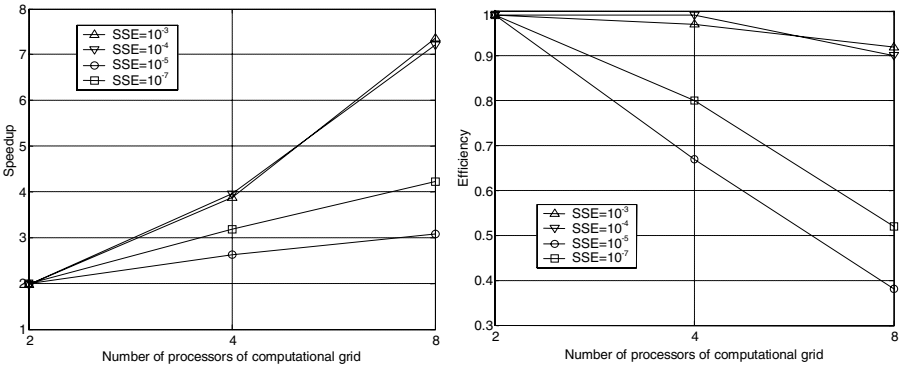


Fig. 3. Speedup and efficiency of parallel algorithm execution on computational grid

Table 3. Efficiency of computational grid vs. parallel computer

CPUs	2	4	8
SSE=10^{-3}	0.02	0.02	0.01
SSE=10^{-4}	0.01	0.03	0.04
SSE=10^{-5}	0.01	-0.01	0.02
SSE=10^{-7}	0.02	-0.12	0.03
Average:		0.0067	

In this connection a communication time (latency time) among processors of the computational grid practically does not influence on the parallelization efficiency despite the fact that it is bigger in 10 times at least in comparison with a communication time among processors of the parallel computer. In particular, experiments has showed, that latency time among two processors is about 0.80 seconds at receiving by *MPI_Recv()* 1000 messages sent by *MPI_Send()* in case of parallel computer Origin300 with NUMAlink interface. This latency time in case of computational grid described above was about 13.31 seconds.

The results of latencies analysis show, that this situation is applicable for coarse-grain parallelization. In case of fine-grain parallelization the calculation time of the code portion could be commensurable with the time of message passing among two processors. Therefore this hardware configuration might not show the same results for other types of parallelizable problems.

5 Conclusions and Future Researches

Usage of computational grids is very promising technology of parallel data processing in modern computational world. In this paper we showed that usage of the computational grid with Globus middleware is more efficient for coarse-graining parallel algorithms of neural networks training in comparison with the parallel computer application. This conclusion is based on the fact that the computational grid showed at least in 4 times bigger performance in comparison with parallel computer, however both parallel systems showed practically the same parallelization efficiency at less costs of the computational grid at least in ten times. In the future works the author plans to investigate an efficiency of fine-grain parallelization approaches of neural networks training on the computational grid with Globus middleware.

Acknowledgement

This work is accomplished within the European INTAS (www.intas.be) Postdoctoral Fellowship grant YSF 03-55-2493 “Development of Parallel Neural Networks Training Algorithms on Advanced High Performance Systems”. This support is gratefully acknowledged.

References

1. Haykin, S.: Neural Networks. Prentice Hall, New Jersey (1999)
2. Mahapatra, S., Mahapatra, R., Chatterji, B.: A parallel formulation of back-propagation learning on distributed memory multiprocessors. Parallel Comp. 22 (12) (1997) 1661-1675
3. Hanzálek, Z.: A parallel algorithm for gradient training of feed-forward neural networks. Parallel Computing. 24 (5-6) (1998) 823-839
4. Chang, L.-C., Chang, F.-J.: An efficient parallel algorithm for LISSOM neural network. Parallel Computing. 28 (11) (2002) 1611-1633
5. Estévez, P. A., Paugam-Moisy, H., Puzenat, D. et al.: A scalable parallel algorithm for training a hierarchical mixture of neural experts. Parallel Computing. 28 (6) (2002) 861-891
6. Murre, J.M.J.: Transputers and neural networks: An analysis of implementation constraints and performance. IEEE Transactions on Neural Networks. 4 (2) (1993) 284–292
7. So, J.J.E., Downar, T.J., Janardhan, R. et al.: Mapping conjugate gradient algorithms for neutron diffusion: applications onto SIMD, MIMD, and mixed-mode machines. International Journal of Parallel Programming. 26 (2) (1998) 183–207
8. Sudhakar, V., Siva Ram Murthy, C.: Efficient mapping of backpropagation algorithm onto a network of workstations. IEEE Trans. on Syst., Man and Cyber. B 28 (6) (1998) 841–848
9. Dongarra, J., Shimasaki, M., Tourancheau, B.: Clusters and computational grids for scientific computing. Parallel Computing. 27 (11) (2001) 1401-1402

10. Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the Grid: enabling scalable virtual organizations. International Journal of Supercomputer Application. 15 (3) (2001)
11. Laforenza, D.: Grid programming: some indications where we are headed. Parallel Computing. 28 (12) (2002) 1733-1752
12. Barberou, N., Garbey, M., Hess, M., Resch, M., Rossi, T., Toivanen, J.: Efficient metacomputing of elliptic linear and non-linear problems. J. Par. Distr. Comp. 63 (5) (2003) 564-577
13. Cooperman, G., Casanova, H., Hayes, J. et al: Using TOP-C and AMPIC to port large parallel applications to the Computation Grid. Fut. Gen. Comp. Sys. 19 (4) (2003) 587-596
14. Takemiya, H., Shudo, K., Tanaka, Y., Sekiguchi, S.: Constructing Grid Applications Using Standard Grid Middleware. Journal of Grid Computing. 1 (2) (2003) 117-131
15. Prodan, R., Fahringer, T.: ZENTURIO: a grid middleware-based tool for experiment management of parallel and distributed applications. J. Par. Distr. Comp. 64 (6) (2004) 693-707
16. Frey, J., Tannenbaum, T., Livny, M., Foster, I., Tuecke, S.: Condor-G: A Computation Management Agent for Multi-Institutional Grids. Cluster Computing. 5 (2002) 237-246
17. Karonis, N.T., Toonen, B., Foster, I.: MPICH-G2: A Grid-enabled implementation of the Message Passing Interface. J. Par. Distr. Comp. 63 (5) (2003) 551-563
18. Foster, I., Kesselman, C.: Globus: a metacomputing infrastructure toolkit, International Journal of Supercomputer Application. 11 (2) (1997) 115–128
19. Pickles, S.M., Brooke, J., Costen, F.C., Gabriel, E., Mueller, M., Resch, M. et al: Metacomputing across intercontinental networks. Fut. Gen. Comp. Sys. 17 (8) (2001) 911–918
20. Turchenko, V.: Parallel Algorithm of Dynamic Mapping of Integrating Historical Data Neural Networks. Information Technologies and Systems. 7 (1) (2004) 45-52
21. Iyengar, S.S.: Distributed Sensor Network - Introduction to the Special Section. Transaction on Systems, Man, and Cybernetics. 21 (5) (1991) 1027-1031
22. Brignell, J.: Digital compensation of sensors. Scient. Instruments. 20 (9) (1987) 1097-1102
23. Sachenko, A., Kochan, V., Turchenko, V., Golovko, V., Savitsky, J., Laopoulos, T.: Method of the training set formation for neural network predicting drift of data acquisition device. Patent #50380. IPC 7 G06F15/18. Ukraine. Filled (04 Jan 2000) Issued (15 Nov 2002) 14
24. Sachenko, A., Kochan, V., Turchenko, V.: Instrumentation for Data Gathering. IEEE Instrumentation and Measurement Magazine. 6 (3) (2003) 34-40
25. Turchenko, V.: Neural network-based methods and means for efficiency improving of distributed sensor data acquisition and processing networks. Ph.D. dissert. National University “Lvivska Politechnika”, Lviv, Ukraine (2001) 188
26. Turchenko, V., Kochan, V., Sachenko, A. and Laopoulos, Th.: Enhanced method of historical data integration using neural networks. Sensors and Systems. 7 (38) (2002) 35-38
27. Happel, B., Murre, J.: Design and evolution of modular neural network architectures. Neural Networks. 7 (1994) 985-1004
28. Petrowski, A., Dreyfus, G., Girault, C.: Performance analysis of a pipelined backpropagation parallel algorithm. IEEE Trans. Neural Networks. 4 (1993) 970–981
29. Paugam-Moisy, H.: Optimal speedup conditions for a parallel back-propagation algorithm. Lecture Notes in Computer Science, Vol. 682. Springer-Verlag (1992) 719–724
30. Hopp, H., Prechelt, L.: CuPit-2: A Portable parallel programming language for artificial neural networks. Proc. 15th IMACS World Congress of Scientific Computation Modeling and Applied Mathematics, Vol. 6. Berlin (1997) 493–498
31. Dongarra, J., Laforenza, D., Orlando, S. (eds.): Recent Advances in Parallel Virtual Machine and Message Passing Interface. Lecture Notes in Computer Science, Vol. 2840. Springer-Verlag, Berlin (2003) ISBN 3-540-20149-1

Object-Oriented Wrapper for Relational Databases in the Data Grid Architecture

Kamil Kuliberda[1], Jacek Wislicki[1], Radoslaw Adamus[1], and Kazimierz Subieta[1,2,3]

[1] Computer Engineering Department, Technical University of Lodz, Lodz, Poland
[2] Institute of Computer Science PAS, Warsaw, Poland
[3] Polish-Japanese Institute of Information Technology, Warsaw, Poland
{kkulibe, jacenty, radamus}@kis.p.lodz.pl
edgar.glowacki@pjwstk.edu.pl, subieta@pjwstk.edu.pl

Abstract. The paper presents a solution of the problem of wrapping relational databases to an object-oriented business model in the data grid architecture. The main problem with this kind of wrappers is how to utilize the native SQL query optimizer, which in majority of RDBMS is transparent for the users. In our solution we use the stack-based approach to query languages, its query language SBQL, updateable object-oriented virtual views and the query modification technique. The architecture rewrites the front-end OO query to a semantically equivalent back-end query addressing the M0 object model that is 1:1 compatible with the relational model. Then, in the resulting SBQL query the wrapper looks for patterns that correspond to optimizable SQL queries. Such patterns are then substituted by dynamic SQL *execute immediately* statements. The method is illustrated by a sufficiently sophisticated example. The method is currently being implemented within the prototype OO server ODRA devoted to Web and grid applications.

1 Introduction

The art of object-oriented wrappers build on top of relational database systems has been developed for years – first papers on the topic are dated to late 80-ties and were devoted to federated databases. The motivation for the wrappers is reducing the technical and cultural difference between traditional relational databases and novel technologies based on object-oriented paradigms, including analysis and design methodologies (e.g. based on UML), object-oriented programming languages (C++, Java, C#, and others), object-oriented middleware (e.g. based on CORBA), object-relational databases and pure object-oriented databases. Recently, Web technologies based on XML/RDF also require similar wrappers. Despite the big pressure on object-oriented and XML-oriented technologies, people are quite happy with relational databases and there is a little probability that the market will massively change soon to other data store paradigms.

Unfortunately, the object-orientedness has as many faces as existing systems, languages and technologies. Thus, the number of combinations of object-oriented options with relational systems and applications is very large. Additionally, wrappers can have different properties, in particular, can be proprietary to applications or generic, can deal with updates or be read-only, can materialize objects on the wrapper

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 367 – 376, 2005.
© Springer-Verlag Berlin Heidelberg 2005

side or deliver purely virtual objects, can deal with object-oriented query language or provide some iterative “one-object-in-a-time” API, etc [1]. This causes an extremely huge number of various ideas and technologies. For instance, Google reports more than 100 000 Web pages as a response to the query “object relational wrapper”.

In this paper we deal with object-to-relational wrappers for distributed, heterogeneous and redundant data and service resources that are to be virtually integrated into a centralized, homogeneous and non-redundant whole. The technology is recently referred to as a “data-intensive grid” or a ”data grid”. While originally the grid technology denotes massive computations that have to be done in parallel on hundreds or thousands of small computers, in business applications a data grid means higher forms of distribution transparency plus some common infrastructures build on top of the grid, including the trust infrastructure (security, privacy, licensing, payments, etc.), web services, distributed transactions, workflow management, etc [2].

The major problem with the described architecture concerns how to utilize an SQL optimizer. In all known RDBMS-s the optimizer and its particular structures (e.g. indices) are transparent to the SQL users. A naive implementation of the wrapper causes that it generates primitive SQL queries such as *select * from R*, and then, processes the results of such queries by SQL cursors. Hence the SQL optimizer has no chances to work. Our experience has shown that direct translation of object-oriented queries into SQL is unfeasible for a sufficiently general case.

The solution to this problem presented in this paper is based on the object-oriented query language SBQL, virtual object-oriented views defined in SBQL, query modification [13], and an architecture that will be able to detect in a query syntactic tree some patterns that can be directly mapped as optimizable SQL queries. The patterns match typical optimization methods that are used by the SQL query optimizer, in particular, indices and fast joins. The idea is currently being implemented within our object-oriented platform ODRA.

The rest of the paper is organized as follows. In Section 2 we present a more detailed discussion concerning object-oriented wrappers built on top of relational databases, including our experience. Section 3 shortly introduces the Stack-Based Approach (SBA) to object-oriented query languages, its query language SBQL and virtual updateable object-oriented views. The section presents only basic ideas - the approach has already resulted in extensive literature (e.g. [12]) and several implementations. Section 4 presents the data grid architecture. Section 5 discusses an object-relational wrapper and presents a simple example showing how it works. Section 6 concludes.

2 More Discussion of the Problem

Mapping between a relational database and a target global object-oriented database should not involve materialization of objects on the global side, i.e. objects delivered by such a wrapper should be virtual. Materialization is simple, but leads to many problems, such as storage capacity, network traffic overhead, synchronization of global objects after updates on local servers, and (for some applications) synchronization of local servers after updates of global objects. Materialization can also be forbidden by security and privacy regulations.

If global objects have to be virtual, they are to be processed by a query language and the wrapper has to be generic, we are coming to concept of virtual object-oriented database views that do the mapping from tables into objects. Till now, however, sufficiently powerful object-oriented views are still a dream, despite a lot of papers and some implementations. The ODMG standard does not even mention views[1]. The SQL-99 standard deals a lot with views, but currently it is perceived as a huge set of loose recommendations rather than as entirely implementable artifact. In our opinion, the Stack-Based Approach and its query language SBQL offer the first and universal solution to the problem of updateable object-oriented database views. In this paper we show that the query language and its view capability can be efficiently used to build optimized object-oriented wrappers on top of relational databases.

Basing on the knowledge and experience[2] gained from our previous attempts to wrap relational content into its object-oriented representation, currently we are implementing (under .NET) an object-oriented platform named ODRA for Web and grid applications, thus the problem of a wrapper on top of relational databases comes again into the play. After previous experience we have made the following conclusions:

- the system will be based on our own, already implemented, object-oriented query language SBQL (described shortly in Section 3), which has many advantages over OQL, XQuery, SQL-99 and other languages,
- the system will be equipped with a powerful mechanism of object-oriented virtual updateable views based on SBQL. Our views have the power of algorithmic programming languages, hence are much more powerful than e.g. SQL views. A partial implementation of SBQL views is ready too [7].

The architecture assumes that a relational database will be seen as a simple object-oriented database, where each tuple of a relation is mapped virtually to a primitive object. Then, on such a database we define object-oriented views that convert such primitive virtual objects into complex, hierarchical virtual objects conforming to the global canonical schema, perhaps with complex repeated attributes and virtual links among the objects. Because SBQL views are algorithmically complete, we are sure that every such a mapping can be expressed. Moreover, because SBQL views can possess a state, have side effects and be connected to classes, one would be able to implement a behavior related to the objects on the SBQL side.

The major problem concerns how to utilize the SQL optimizer. After our previous experience we have concluded that static (compile time) mapping of SBQL queries into SQL is unfeasible. On the other hand, a naive implementation of the wrapper, as presented above, leaves no chances to the SQL optimizer. Hence we must use optimizable SQL queries on the back-end of the wrapper.

The solution of this problem is presented in this paper. It combines SBQL query engine with the SQL query engine. There are a lot of various methods used by an SQL optimizer, but we can focus on three major ones: *rewriting* (e.g. pushing selections before joins), *indices* (i.e. internal auxiliary structures for a fast access), *fast joins* (e.g. hash joins).

---

[1] The *define* clause of OQL is claimed to be a view, but this is misunderstanding: it is a macro-definition (a textual shorthand) on the client-side, while views are server-side entities.

[2] A gateway from the DBPL system to Ingres and Oracle (1993) and a part of the European project ICONS (Intelligent COntent maNagement System), IST-2001-32429.

Concerning rewriting, our methods are perhaps as good as SQL ones, thus this kind of optimization will be done on the SBQL side. Two next optimizations cannot be done on the SBQL side. The idea is that an SBQL syntactic query tree is first modified by views [13], thus we obtain a much larger tree, but addressing a primitive object database that is 1:1 mapping of the corresponding relational databases. Then, in the resulting tree we are looking for some patterns that can be mapped to SQL and which enforce SQL to use its optimization method. For instance, if we know that the relational database has an index for Names of Persons, we are looking in the tree the sub-trees representing the SBQL query such as:

```
Person where Name = "Doe"
```

After finding such a pattern we substitute it by the dynamic SQL statement:

```
exec_immediately(select * from Person where Name = "Doe")
```

enforcing SQL to use the index. The result returned by the statement is converted to the SBQL format. Similarly for other optimization cases. In effect, we do not require that the entire SBQL query syntactic is to be translated to SQL. We interpret the tree as usual by the SBQL engine, with except of some places, where instead of some subtrees we issue SQL *execute immediately* statements.

3 Stack Based Approach, SBQL and Updatable Object Views

In the stack-based approach (SBA) a query language is considered a special kind of a programming language. Thus, the semantics of queries is based on mechanisms well known from programming languages like the environment stack. SBA extends this concept for the case of query operators (selection, projection/navigation, join, quantifiers, etc.). Using SBA, one is able to determine precisely the operational semantics (abstract implementation) of query languages, including relationships with object-oriented concepts, embedding queries into imperative constructs, and embedding queries into programming abstractions: procedures, functional procedures, views, methods, modules, etc.

SBA is defined for a general object store model. Because various object models introduce a lot of incompatible notions, SBA assumes some families of object store models which are enumerated M0, M1, M2 and M3. The simplest is M0, which covers relational, nested-relational and XML-oriented databases. M0 assumes hierarchical objects with no limitations concerning nesting of objects and collections. M0 covers also binary links (relationships) between objects. Higher-level store models introduce classes and static inheritance (M1), object roles and dynamic inheritance (M2), and encapsulation (M3). For these models there have been defined and implemented the query language SBQL, which is much more powerful than ODMG OQL [10] and XML-oriented query languages such as XQuery [14]. SBQL, together with imperative extensions and abstractions, has the computational power of programming languages, similarly to Oracle PL/SQL or SQL-99.

Rigorous formal semantics implied by SBA creates a very high potential for the query optimization. Several optimization methods have been developed and implemented, in particular methods based on query rewriting, indices, removing dead queries, and others [11].

SBQL is based on the principle of compositionality, i.e. semantics of a complex query is recursively built from semantics of its components. In SBQL, each binary operator is either algebraic or non-algebraic. Examples of algebraic operators are numerical and string operators and comparisons, aggregate functions, union, etc. Examples of non-algebraic operators are selection (where), projection/navigation (the dot), join, quantifiers (∃, ∀), and transitive closures. The semantics of non-algebraic operators is based on a classical environmental stack, thus the name of the approach.

The idea of SBQL updatable views relies in augmenting the definition of a view with the information on user intentions with respect to updating operations. The first part of the definition of a view is the function, which maps stored objects onto virtual objects (similarly to SQL), while the second part contains redefinitions of generic operations on virtual objects. The definition of a view usually contains definitions of subviews, which are defined by the same principle [4].

The first part of the definition of a view has the form of a functional procedure. It returns entities called *seeds* that unambiguously identify virtual objects (usually seeds are OIDs of stored objects). Seeds are then (implicitly) passed as parameters of procedures that overload operations on virtual objects. These operations are determined in the second part of the definition of the view. There are distinguished several generic operations that can be performed on virtual objects: *delete* removes the given virtual object, *retrieve* (dereference) returns the value of the given virtual object, *navigate* navigates according to the given virtual pointer, *update* modifies the value of the given virtual object according to a parameter, etc.

All procedures, including the function supplying seeds of virtual objects are defined in SBQL and can be arbitrarily complex [4].

4 Architecture of the Data Grid

Figure 1 shows the architecture of a data grid. Its central part is the *global virtual store* containing virtual objects and services. Its role is to store addresses of local

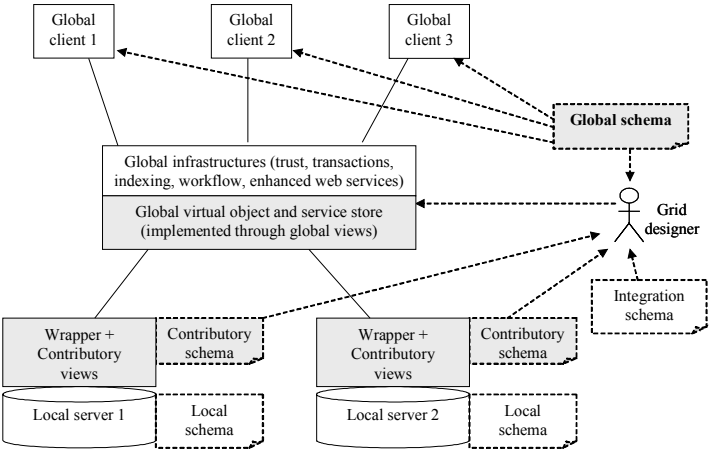


Fig. 1. Architecture of a data grid

servers and to process queries sent from *global client* applications. The global virtual store presents the business objects and services according to the *global schema*, which has to be defined and agreed upon the organization creating the grid. The global schema is used by programmers to create global client applications. The grid integrates services and objects physically stored in the *local servers*. Administrators of local servers define *contributory schemata* and corresponding *contributory views* [3, 5], mapping local data and services to the global schema demands. Local data can be stored within any kind of DBMS providing a corresponding wrapper plus contributory views are implemented. The *global virtual store* is a collection of views that are responsible for the integration of distributed, heterogeneous and redundant resources and ensure higher-level transparencies. The contributory views and global views are updatable. The *integration schema* presents information on dependencies between local servers (replications, redundancies, etc.) [3, 6].

5 Architecture of the Object-Relational Wrapper and Examples

Figure 2 presents the architecture of the wrapper. The general assumptions are the following:

- externally the data are designed according to the OO model and the business intention of the *global schema* – the *front-end* of the wrapper (SBQL),
- internally the relational structures are presented in the M0 model (excluding pointers and nesting levels above 2) [12] – the *back-end* of the wrapper (SBQL),
- the mappings between *front-end* and *back-end* is defined with updatable object views. They role is to map *back-end* into *front-end* for querying and *front-end* onto *back-end* for updating (virtual objects),
- for global queries, if some not very strict conditions are satisfied, the mapping form front-end into back-end query trees is done through query modification, i.e macro-substituting every view invocations in a query by the view body.

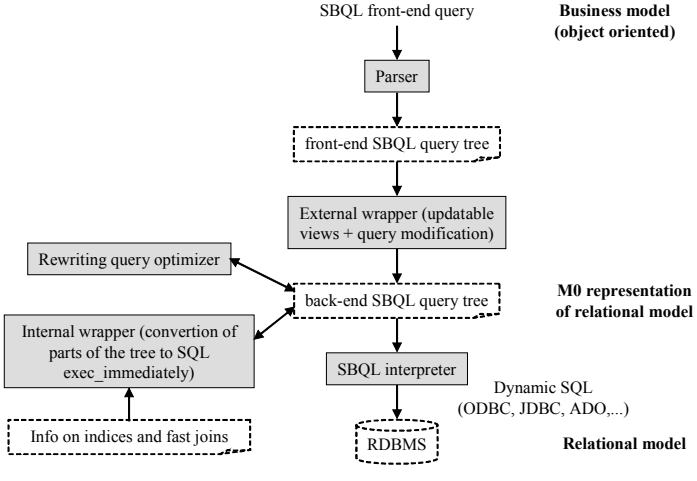


Fig.2. The architecture of a generic wrapper for relational databases

5.1 Updates Through the Wrapper and the Optimization Procedure

The presented architecture assumes retrieval operations only, because the query modification technique assumed in this architecture does not work for updates. However, the situation is not hopeless (although more challenging). Because in SBQL updates are parameterized by queries, the major optimizations concern just these parameters, with the use of the query modification technique as well. Then, after the optimization, we can develop algorithms that would recognize in the back-end query tree all updating operations and then, would attempt to change them to dynamic SQL *update*, *delete* and *insert* statements. There are technical problems with identification of relational tuple within the SBQL engine (and further in SQL). Not all relational systems support *tuple identifiers* (tids). If tids are not supported, the developers of a wrappers must relay on a combination (*relation_name, primary_key_value(s)*), which is much more complicated in implementation. Tids (supported by SQL) simply and completely solve the problem of any kind of updates.

In Figure 2 we have assumed that the internal wrapper utilizes information on indices and fast joins (primary-foreign key dependencies) available in the given RDBMS. In cases of some RDBMS (e.g. MS SQL Server) this information cannot be derived from the catalogs. Then, the developers are forced to provide an utility allowing the wrapper designer to introduce this information manually.

The query optimization procedure (looking from wrapper's front-end to back-end) for the proposed solution can be divided into several steps:

1. Query modification is applied to all view invocations in a query, which are macro-substituted with seed definitions of the views. If an invocation is preceded by the dereference operator, instead of the seed definition, the corresponding *on_retrieve* function is used (analogically, *on_navigate* for virtual pointers). The effect is a monster huge SBQL query referring to the M0 version of the relational model available at the back-end.
2. The query is rewritten according to static optimization methods defined for SBQL [11] such as removing dead sub-queries, factoring out independent sub-queries, pushing expensive operators (e.g. joins) down in the syntax tree, etc. The resulting query is SBQL-optimized, but still no SQL optimization is applied.
3. According to the available information about the SQL optimizer, the back-end wrapper's mechanisms analyze the SBQL query in order to recognize patterns representing SQL-optimizable queries. Then, *exec_immediately* clauses are issued.
4. The results returned by *exec_immediately* are pushed onto the SBQL result stack as collections of structures, which are then used for regular SBQL query evaluation.

5.2 Optimization Example

The example discusses a simple two-table relational database containing information about employees `EmpR` and departments `DeptR`, “R” stands for “relational” to increase the clearness (fig. 3).

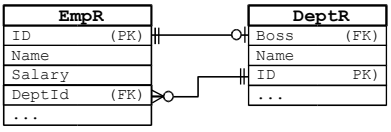


Fig.3. The example of a relational schema

The relational schema is wrapped into an object schema shown in figure 4 according to the following view definitions. The `EmpR-DeptR` relationship is realized with `worksIn` and `boss` virtual pointers:

```
create view EmpDef {
  virtual_objects Emp {return EmpR as e;}
  virtual_objects Emp(EmpId) {return (EmpR where ID == EmpId) as e;}
  create view nameDef {
    virtual_objects name{return e.name as n;}
    on_retrieve {return n;}
  }
  create view salaryDef {
    virtual_objects salary {return e.salary as s;}
    on_retrieve {return s;}
  }
  create view worksInDef {
    virtual_pointers worksIn {return e.deptID as w;}
    on_navigate {return Dept(w) as Dept;}
  }
}

create view DeptDef {
  virtual_objects Dept {return DeptR as d;}
  virtual_objects Dept(DeptId) {return (DeptR where ID == DeptId) as d;}
  create view nameDef {
    virtual_objects name {return d.name as n;}
    on_retrieve {return n;}
  }
  create view bossDef {
    virtual_pointers boss {return e.bossID as b;}
    on_navigate {return Emp(b) as Emp;}
  }
}
```

Fig. 4. Object schema used in the optimization example (wrapper's front-end)

Consider a query appearing at the front-end (visible as a business database schema) that aims to *retrieve names of the employees working in the "Retail" department with salary the same as the employee named Doe's*. The query can be formulated as follows (we assume that there is only one employee with that name in the store):

```
((Emp where worksIn.Dept.name == "Retail") where
   salary == ((Emp where name == "Doe").salary)).name;
```

The information about the local schema (the relational model) available to the wrapper that can be used during query optimization is that the `name` column is uniquely indexed and there is a primary-foreign key integrity between `DeptId` column (`EmpR` table) and `ID` column (`DeptR` table).

The optimization procedure is performed in the following steps:

1. Introduce implicit `deref` (dereference) functions
```
((Emp where worksIn.Dept.deref(name) == "Retail") where deref(salary)
== (Emp where deref(name) == "Doe").deref(salary)).deref(name);
```
2. Substitute `deref` with the invocation of `on_retrieve` function for virtual objects and `on_navigate` for virtual pointers
```
((Emp where worksIn.(Dept(w) as Dept).Dept.(name.n) == "Retail")
where (salary.s) == (Emp where (name.n) == Doe").(salary.s)).(name.n);
```
3. Substitute all view invocations with the queries from *sack* definitions
```
(((EmpR as e) where ((e.deptID as w).(((DeptR where ID == w) as d) as
Dept)).Dept.((d.name as n).n) == "Retail") where ((e.salary as s).s)
== ((EmpR as e) where ((e.name as n).n) == "Doe").((e.salary as
s).s)).((e.name as n).n);
```
4. Remove auxiliary names `s` and `n`
```
(((EmpR as e) where ((e.deptID as w).(((DeptR where ID == w) as d) as
Dept)).Dept.(d.name) == "Retail") where (e.salary) == ((EmpR as e)
where (e.name) == "Doe").(e.salary)).(e.name);
```
5. Remove auxiliary names `e` and `d`
```
((EmpR where ((deptID as w).((DeptR where ID == w) as Dept)).Dept.name
== "Retail") where salary == (EmpR where name == "Doe").salary).name;
```
6. Remove auxiliary names `w` and `Dept`
```
((EmpR where (DeptR where ID == deptID ).name == "Retail") where
salary == (EmpR  where name == "Doe").salary).name;
```
7. Now take common part before loop to prevent multiple evaluation of a query calculating salary value for Emp named *Doe*
```
((((EmpR where name == "Doe").salary ) group as z).(EmpR where
((DeptR where ID == deptID).name == "Retail")) where salary == z).name;
```
8. Connect where and navigation clause into one where connected with and operator
```
((((EmpR where name == "Doe").salary ) group as z).(EmpR where (DeptR
where (ID == deptID and name == "Retail")) where salary == z).name;
```
9. Because `name` column is uniquely indexed, the sub-query `(EmpR  where name == "Doe")` can be substituted with *exec_ immediately* clause
```
(((exec_immediately("SELECT salary FROM EmpR WHERE name = 'Doe'"))
group as z).(EmpR where (DeptR where (ID == deptID and name ==
"Retail")) where salary == z).name;
```
10. Because the integrity constraint with `EmpR.DeptId` column and `DeptR.ID` column is available to the wrapper, the pattern is detected and another *exec_immediately* substitution is performed:
```
(((exec_immediately("SELECT salary FROM EmpR WHERE name = 'Doe'"))
group as z).(exec_immediately("SELECT * FROM EmpR, DeptR WHERE
EmpR.deptID = DeptR.ID AND DeptR.name = 'Retail'") where salary ==
z).name;
```

Either of the SQL queries invoked by *exec_immediately* clause is executed in the local relational resource and pends native optimization procedures (with application of indices and fast join, respectively).

6 Conclusions

We have presented the approach to wrapping relational databases to an object-oriented business model. The approach assumes the stack-based approach, its query language SBQL, updatable views and the query modification technique. As shown in

the example, a front-end SBQL query can be modified and optimized with application of SBA rules and updatable views within the wrapper and then the native relational optimizers for SQL language can be employed. The described wrapper architecture enables building generic solutions allowing presentation of data stored in various relational resources as object-oriented models visible at the top level of the grid and accessing the data with object query language.

The described optimization process assumes correct relational-to-object model transformation (with no loss of database logic) and accessibility of the relational model optimization information such as indices and/or primary-foreign key relations.

The method is currently being implemented as a part of our new project ODRA devoted to Web and grid applications.

References

1. Bergamaschi, S., Garuti, A., Sartori, C., Venuta, A.: Object Wrapper: An Object-Oriented Interface for Relational Databases. EUROMICRO 1997, pp.41-46
2. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Global Grid Forum, June 22, 2002
3. Kaczmarski, K., Habela, P., Subieta, K.: Metadata in a Data Grid Construction. Proc. of 13th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE-2004), Italy, June, 2004
4. Kozakiewicz, H., Leszczyłowski, J., Płodzień, J., Subieta, K.: Updateable Object Views. ICS PAS Reports 950, October 2002
5. Kozankiewicz, H., Stencel, K., Subieta, K.: Implementation of Federated Databases through Updateable Views. Proc. EGC 2005 - European Grid Conference, Springer LNCS, 2005, to appear
6. Kozankiewicz, H., Stencel, K., Subieta, K.: Integration of Heterogeneous Resources through Updatable Views. Workshop on Emerging Technologies for Next generation GRID (ETNGRID-2004), 13th IEEE WETICE-2004, University of Modena and Reggio Emilia, Italy, June 14-16, 2004, Proceedings published by IEEE
7. Kozankiewicz, H., Subieta, K.: SBQL Views - Prototype of Updateable Views. ADBIS (Local Proceedings) 2004
8. Matthes, F., Rudloff A., Schmidt, J.W., Subieta, K.: A Gateway from DBPL to Ingres. Proc. of Intl. Conf. on Applications of Databases, Vadstena, Sweden, Springer LNCS 819, pp.365-380, 1994
9. Moore, R., Merzky, A.: Persistent Archive Concepts. Global Grid Forum GFD-I.026. December-2003
10. Object Data Management Group: The Object Database Standard ODMG, Release 3.0. R.G.G.Cattel, D.K.Barry, Ed., Morgan Kaufmann, 2000
11. Plodzien, J.: Optimization Methods in Object Query Languages, PhD Thesis. IPIPAN, Warszawa 2000
12. Subieta, K.: Theory and Construction of Object-Oriented Query Languages. Editors of the Polish-Japanese Institute of Information Technology, 2004, 522 pages
13. Subieta, K., Plodzien, J.: Object Views and Query Modification, (in) Databases and Information Systems (eds. J. Barzdins, A. Caplinskas), Kluwer Academic Publishers, pp. 3-14, 2001
14. W3C: XQuery 1.0: An XML Query Language. W3C Working Draft 12, November 2003, http://www.w3.org/TR/xquery/

Modeling Object Views in Distributed Query Processing on the Grid

Krzysztof Kaczmarski[1], Piotr Habela[2], Hanna Kozankiewicz[3], and Kazimierz Subieta[2,3]

[1] Warsaw University of Technology, Warsaw, Poland
k.kaczmarski@mini.pw.edu.pl
[2] Polish-Japanese Institute of Information Technology, Warsaw, Poland
habela@pjwstk.edu.pl
[3] Institute of Computer Science PAS, Warsaw, Poland
{hanka, subieta}@ipipan.waw.pl

Abstract. This paper proposes a method for modeling views in Grid databases. Views are understood as independent data transformation services that may be integrated with other Database Grid Services. We show examples of graphical notation and semi-automated query construction. The solution is supported by a minimal metamodel, which also provides reflection capabilities for dynamic services orchestration.

1 Introduction

The Grid applications, integrating the whole variety of different computer systems recently became extremely complex. Emerging OGSA standard is a significant improvement of distributed systems' integration. With the introduction of Grid Database Services, OGSA-DAI and OGSA-DQP, access to distributed database resources, querying, optimizations, parallelization and analysis has been greatly simplified [10]. In services-based philosophy users formulate queries and send them to dynamically created Grid Data Service, which is responsible for the rest of the job, and which may use other services to achieve results in the best possible way, for example a Distributed Query Service combined with several Grid Query Evaluation Services. However, formulating queries directly over distributed data, having in mind all possible fragmentations is very difficult if not impossible for most users. What we need in distributed systems, especially in heterogeneous environments, are higher-level interfaces, which could transparently integrate fragmented data and transform it to our needs.

Here, we propose updatable object views as a convenient means of defining higher abstraction over tangled and distributed data. OGSA-DAI defines two general kinds of components: *data access components* and *data integration components*. Our updatable views are rather the latter case, but could also be called *data transformation components,* since they do not store any data and are not limited to any particular task like integration. The advantage of our solution is its simplicity and minimalism beneficial for Grid end users, who are not aware of the local databases heterogeneity, neither data fragmentation nor other aspects of data distribution.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 377 – 386, 2005.
© Springer-Verlag Berlin Heidelberg 2005

1.1 Motivation and Related Work

Our main motivation is to provide a support for development process of such a Grid database, in which certain nodes have a role of data transformation points. We see each view as an object's interface transformation service, which has its semantics, data input and output. Such a data transformation component may be wrapped by a dedicated Data Grid Service and used among other data services, serving high-level integrated data. What is more important is that such a wrapped data transformation component may be used by DQP services in many places according to an optimal query evaluation plan. The only limitation is that our updatable view must by created by a database designer in order to specify the semantics of generic update operations performed through the view. The goal is to propose a modeling tool supporting such object transformations and their compositions. The designer's tasks are describing resources consumed or produced by certain views; and finding and describing dependencies between consumed and produced objects, which in fact describe transformations' logics. To assist these tasks, in the rest of the paper we focus on:

- modeling virtual and real objects' interfaces as descriptions of views' input and output
- describing sequences of transformations, tracking dependencies between interfaces
- recognizing basic patterns of transformations performed by a view
- establishing constraints between views.

One of the recent approaches to visual modeling of database views or virtual objects except for the simple view notation in UML for database design [9]. Our notation is much richer and dedicated to distributed database systems.

There are also many multiparadigmatic database query building proposals based on forms, graphical notations or virtual reality [13]. However, they are focused rather on inexperienced database users, while we support database designers and integrators.

Data transformations (integrations) performed by specialized views, creating virtual resources, are common in many systems. In distributed databases, there are two major, well-known approaches to integrate data: global-as-view [1], where the data in the global schema are defined as views over the data in the sources, and local-as-view, where the data in the sources are defined as views over the global schema [8]. In this paper, we present a mutation of global-as-view approach, which creates virtual data in a form consumed by users, but distributed among nodes and ready not only to serve local but also global users. Piazza Peer Data Management System [12] uses similar way of exchanging information between nodes. In Piazza, schema mediation may be performed dynamically, plus peers may attach and detach dynamically. We consider allowing such a possibility in updatable views, based on the mechanisms located higher in meta-levels' hierarchy and allowing generic programming over metadata. This topic is, however, beyond the scope of this paper.

2 Views in a Grid Database

A Grid Database, created and used by a consortium [2], consists of independent database systems, and specialized Grid Services, which share data i.e., publish and

consume objects. Each of them may use two kinds of specialized views to achieve desired purposes:

Contributory view (sometimes called *mediator*) is a view by means of which a node shares its own resources with the others. Its main task is to hide heterogeneity of local database systems (object, relational, etc.) within the consortium, by transforming local data models into unified data model specific for the consortium. Its second task is controlling access rights and hiding data, which should not be published.

Grid view is a view performing a Grid Data Transformation Service for users (local or global). Through this view, a user sees Grid resources adapted to his/her particular needs. All data transformations, like integration, are transparent. Users see only resulting objects created according to a predefined schema. We say that data created by the view is *virtual,* because it does not exist in any concrete place but is created on the fly, when needed upon distributed resources [5].

One of the most important features of our approach to data transformation in Grid is that virtual objects are indistinguishable from real objects. Applications may use data from local repository exactly in the same way as data produced by Grid views. In this way, transparency of data fragmentation and transparency of location are satisfied. However, if a Grid view uses data provided by other views in the system, it never knows where data come from and where exactly a query will be executed. The request may propagate over the network [4]. Thus, a virtual object created by a Grid view indirectly depends on the whole *sequence of transformations.* Describing and analyzing possible views settings is one of the database designer's tasks.

2.1 A Generic Data Transformation Based on an Updatable View

A mature view-based database system has to support not only data retrieval but also data updates. Classical database views (materialized or not) have limited functionality in this field. Our data model and query language allows to specify the intended virtual data update behavior through dedicated view's procedures.

The view mechanism is defined in terms of the Stack-Based Approach (SBA) [11], which assumes that query languages are a special kind of programming languages and can be formallized in a similar manner (Stack-Based Query Language). A database view definition is not a single query (as in SQL), but it is a complex structure. It consists of two parts: the first one determines the so-called *seeds* (the values or references to stored objects that are the basis for building up virtual objects), and the second one redefines the generic operations on virtual objects [5]. The first part of the view definition is an arbitrarily complex functional procedure, and is similar to extraction programs used in [7]. The *seeds* it returns are passed as parameters for the operations on virtual objects. The operations have the form of procedures that override default updating operations. We identified four generic operations that can be performed on virtual objects (which also completely cover CRUD functionality):

Updating, which assigns a new value to the virtual object. A parameter the procedure accepts is the new value to be assigned.

Deletion, which deletes the virtual object.

Insertion, which inserts a new object into the given virtual object.

Dereference, which returns the value of the given virtual object.

For a given view, an arbitrary subset of these operations can be defined. If any operation is not defined, it means it is forbidden (we assume no updating through side effects, e.g. by references returned by a view invocation).

Moreover, a view definition may contain nested views, defined within the containing view's environment. Thus, arbitrarily nested complex objects may be constructed.

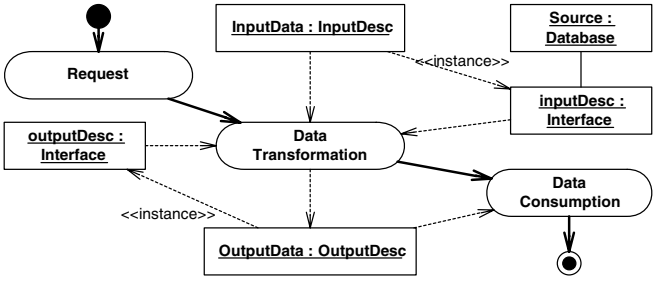


Fig. 1. A generic data transformation performed by a single view uses (meta) description of input and output data (maybe implicit, known only to programmer), collections of input data and produces output data in desired form. Input description is connected to certain location, which is separate information and thus may be easily changed.

When a view is invoked in a query, it returns a set of virtual identifiers (that are counterparts of the identifiers of stored objects). Next, when a system tries to perform update operation with a virtual identifier as an l-value, it recognizes that it deals with the virtual object and calls a proper update operation from the view definition. To enable that, a virtual identifier must contain both a seed and the identifier of the view definition. The whole process of view updating is internal to the proposed mechanism and is invisible to view users, who deal with virtual objects in the same manner as with real objects. This feature is known as view *transparency* and it is a key requirement for the view mechanism in our data integration approach.

In views, data transformations are defined using any procedural and declarative constructs allowed in SBQL. Fig. 1 shows a view performing a generic transformation of *InputData* into *OutputData.* The transformation is described by interfaces of input data (*InputDataDesciption)* and interfaces of output data (*OutputDataDescription).* For simplicity in the figure, data transformation consumes and produces only one type of objects, but it is not a requirement.

Each interface transformation has its semantics intended by the designer to produce certain data objects. We may recognize several well-known types of typical data transformations – *transformation patterns*. For example, we may consider hiding some information by selecting only desired object's attributes [3] (*projection pattern*), or aggregate – e.g. a collection of integers to calculate an average value. One of the most important transformation patterns in distributed scenario are merging and joining views.

Please note, that we do not explicitly say here, whether transformation is performed by contributory view, or Grid view. It is not important from conceptual point of view, since the difference is only in source data used. Data distribution is transpar-

ent, so in fact, within assumed privileges, a view may access data anywhere. For modeling views and sequences of transformations, their location is less important.

2.2 Sequences of Transformations

In the mentioned Grid database system, each view may offer virtual objects based on source objects, which may be concrete or virtual. This ability is important for flexibility of the system. Usage of virtual objects by another view builds a sequence of transformations. Data objects may travel through several nodes and undergo transformations before their final consumption. In some cases, it may be important to look for optimizations to minimize transfer and unnecessary transformations. In larger Grid systems, automated DQP Service supported by an optimizer and query evaluator may decide to move some parts of a data transformation sequence to nodes that are more powerful or ones closer to the end user, achieving performance improvements. These systems would have to use meta-information concerning involved objects' interfaces and be able to infer about their compatibility. This inference may sometimes require human cooperation.

3 Transformations Modeling

This section introduces notations for modeling various transformations on data that may appear in a Grid. However, please note that the transition from a design to an implementation, by for instance code generators, cannot be fully automated in case of Grid databases. This is because of two reasons: limitations of sensible graphical modeling notation, which is not capable of providing all the necessary details (except perhaps for the simplest patterns) and independence of nodes in distributed systems. A Grid system may always face problems of missing fragments of objects, contradictory or incomplete information [6]. Thus, creation of updatable view requires manual (or at least partially manual) implementation of two basic parts: virtual object creation (seeds) upon source objects, which must handle all exceptional situations, and virtual object changeability implementation methods, which must react to users' operations. We propose assistance for some parts of implementation tasks, but manual programming or at least human control, would always be necessary.

3.1 Modeling Virtual Objects

As it was explained earlier, describing data objects and dependencies between them is crucial for designing data transformations in Grid databases. Here, we propose solutions for extended interfaces and dependencies descriptions.

Objects Interfaces. Comparing to standard UML notation, externally visible interfaces of virtual and concrete objects in Grid database require additional constructs for proper showing the changeability allowed for particular (virtual or concrete – treated uniformly) objects. Distinction of the composition of nested objects and references among objects, which is important for object databases, can be solved with the standard UML notation, if there is an agreement on the semantics of the composition relationship. However, changeability flags would need the following symbols

(see Fig. 2): *isUpdatable* ("!"); *isDereferenceable* ("?"); *isRemovable* ("^"); *isInsertable* (">"). The symbols can appear before a feature name (or before a class name in the simplified syntax suggested in Fig. 2). The changeability symbols are shown within the curly brackets to allow suppressing changeabilities (by showing no brackets, to distinguish from declaring a feature with none of the changeabilities allowed).

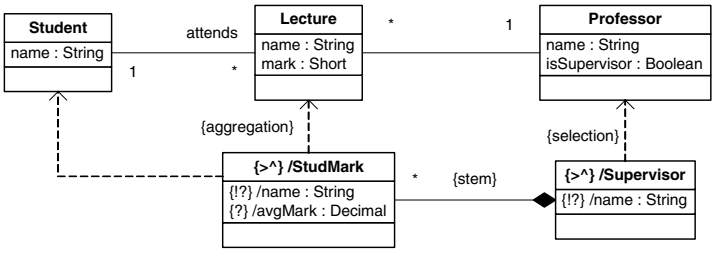


Fig. 2. A sample definition of virtual objects' interfaces (transformations hidden for clarity). Assume the data are restructured according to the needs of some external system (e.g. a statistical analysis subsystem), which should not have any access to the identities of the students. *StudMark* and *Supervisor* show also the changeability notation. Selection means, that to provide Supervisor virtual objects only certain *Professor* source objects are selected. Aggregation indicates that a number of *Lecture* objects are used to create a single *StudMark* object. 'Stem' label indicates preserving the structure (and dependency) of source objects. Here *StudMark* depends on *Supervisor* as *Lecture* is connected to *Professor*.

Dependency Illustration. Here, we suggest notation based on the generic UML dependency relationship and using the same graphical notation (labeled «view dependency» if necessary). Notice that for pragmatic reasons we simplify the notation. Although the view dependencies span between structural features the dependency arrows are drawn rather between their classes. In contrast to the regular dependency arrow, view dependency can additionally indicate (using keywords within curly brackets, as shown in Fig. 2), the selectivity and aggregation property (*selection* and *aggregation* keywords respectively). To indicate that particular complex view (that is, a view containing other views) preserves the structure of its source object (mapping the features of the latter), we use the *stem* keyword in the properties representing subviews.

3.2 Modeling Sequences of Transformations

Simple dependency arrows as shown in the previous section are enough to indicate necessary interface transformation when a single virtual object depends on one source object. A more complicated case, especially data integration, may introduce additional dependencies between sources and results. Fig. 3 shows an example of several data objects correlated by a common join operation. We may notice following properties of this joining transformation:

- All source objects must have a composition key used in join (here, ID)
- Resulting virtual object have at most data from input objects

– If some corresponding part of an object (say scholarship) may be missing it should be reflected in output interface as an optional property. Otherwise output virtual object cannot be created.

However, the diagram says nothing about constraints or situations in which some data are missing. Programmers of updatable views must overcome these problems. In the same way, extraction programs from [7] solve the problem of uncertainness.

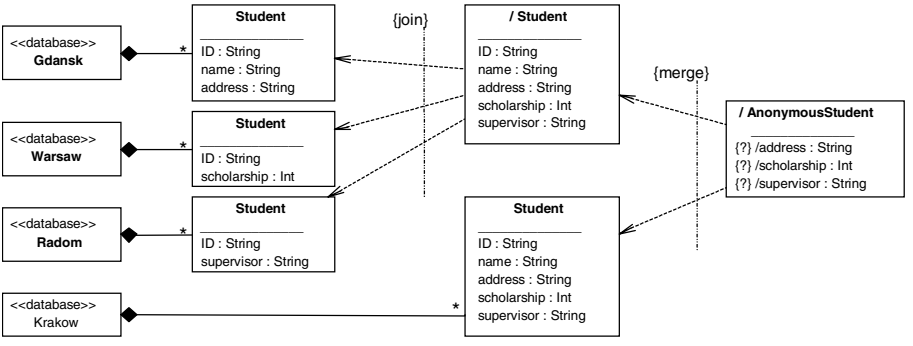


Fig. 3. Example integration of distributed data. Two transformations realize patterns named `join' and 'merge'. For clarity, we do not show explicitly nodes offering virtual objects. Objects' composition is performed using certain keys. Here, it would be student's ID. Please note, that we do not consider expressions defining keys in this paper.

Transformations' compositions may be done only under constraints on input and output data types. In the Fig. 3, merging transformation uses results of the previous join. It unions two collections of data: from Krakow database and the newly joined objects. Union of collections of objects is sensible if all objects have the same semantics, which is expressed by their interfaces. Because interface of Student objects in Krakow is known and fixed, it constrains output interface of the join transformation.

An important feature of diagram in Fig. 3 is that objects' interfaces and thus views, which produce them, are separated from certain locations. As it was quoted earlier, for transformation's semantics data origin is not important. However, it becomes important for system's implementation. Thus, to support transition from the design to the implementation we must supply information about concrete roles of database nodes in data exchange. It means that each interface used in the Grid should be explicitly assigned to a node or be somehow connected (specialization, reference, etc.) to an interface, which is already assigned. If objects implementing same interface are located in many nodes each location must be shown explicitly (like two identical *Student* objects before merging transformation in Fig. 3).

Separation of a data source, transformation's semantics and interfaces simplifies insertion of a new transformation or a data source inside a previously modeled sequence. A new interface or a transformation has to agree with established standardized data description, plus it must be known to the rest of the transformations in the chain, which stay unmodified. Let us consider introducing a new node offering

Student objects similar to those published by Krakow database from Fig. 3. Merging view has only to get the information about the new data source. That could be supplied in the runtime by a dynamical list of data sources or statically by modifying the transformation chain diagram and regenerating code for particular views.

Example Sequence of Transformations. If the semantics of a transformation is of known pattern, then even if updatable view's code generation is not completely automated, a CASE tool could suggest partial solution. Below we present transformations from Fig. 3 implemented in a single query. It uses input and output interfaces definitions and source objects location. Combination of many transformations in a single query is possible because of powerful capabilities of query composition in SBQL.

1. Performing 'join' (transformation's semantics) and shaping the output for the next transformation (output–input constraint):

```
((Gdansk.Student as g) join (Warsaw.Student as w where w.ID=g.ID)
join (Radom.Student as r where r.ID=g.ID)).(g.ID as ID, g.name as
name, g.address as address, w.scholarship as scholarship,
r.supervisor as supervisor)) as Student
```

2. Adding 'merge' (transformation's semantics) and creating final output required by output constraints (output constraint):

```
((((Gdansk.Student as g) join (Warsaw.Student as w where w.ID=g.ID)
join Radom.Student as r where r.ID=g.ID)).(g.ID as ID, g.name as
name, g.address as address, w.scholarship as scholarship,
r.supervisor as supervisor)) as Student) union Cracow.Student as Stu-
dent)).
(Student.address, Student.scholarship, Student.supervisor))
as AnonymousStudent
```

Presented query is only the updatable view's data retrieving declaration. We may see the model's influence on its certain parts. The rest of the view definition, which supports updating, deleting, inserting and dereferencing procedures, has to be programmed separately.

3.3 Metamodel Supporting Transformations

At the metamodel level, we decided to treat all objects' interfaces in the same way, regardless of their virtual or concrete nature. Notions, which are in fact view-related but have influence on how we see resulting objects, are kept in *StructuralFeature* class. The only change into existing UML notions is the replacement of the changeability attribute from that class by four boolean attributes. We use the following names: *isUpdatable*, *isRemovable*, *isInsertable* and *isDereferenceable*. We assume that those structural features, which posses (standard-defined) tagged value "derived" represent virtual objects and may be the subject of data dependency specifications.

Database class represents a source node offering objects. Transformation is attached to certain location though *StructuralFeature*, which describes its output. In other words, a view must be located in a node, which is to offer certain objects.

Transformation class is responsible for describing transformation performed by an updatable view to produce virtual objects, pointed by *produces*. Interfaces consumed by a view are enumerated by multiple *uses* reference. Although it is not possible to precisely describe visually how a given virtual object is computed, some information

can be easily provided concerning the characteristics of a view dependencies and relations between them, which are in fact data integration patterns. Kind of a transformation may be set in *IntPaternKind* (we follow UML style here). The basic transformation patterns concerning data integration are *projection, merge* and *join*, but metamodel is open for any other custom defined and named patterns. The implementation phase uses this description to create certain template for query matching desired semantics.

Dependency between transformation and interfaces it consumes may be additionally described by mutually orthogonal flags: *isSelective* (Source data are used to select only the objects meeting a given criteria), *isAggregating* (This property indicates that a given virtual object realizes a many-to-one mapping of the source data). This description affects the kind of query that is to be used to create seeds of virtual objects.

The proposed transformation properties are not exhaustive, as it is not possible to cover with such notions the whole expressiveness of even the most typical queries that may be used as view definitions. However, it provides some hint concerning the intent of a given view, with the level of detail that is feasible to show on a diagram and enough to constrain and control implementation of a modeled view.

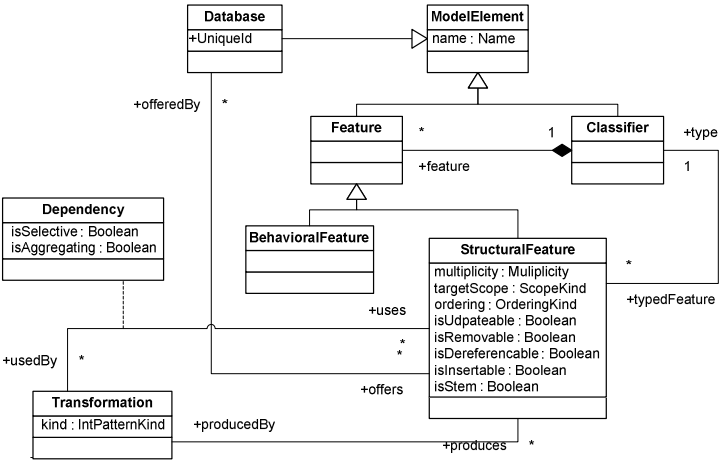


Fig. 4. A fragment of UML Metamodel extended by features necessary for proper dependency tracking between transformations

4 Conclusions and Future Work

In this paper, we have presented a feasible method supporting a user in modeling views in Grid Databases. A Grid designer is provided with tools to model data transformation and to constrain transformation's chains. Our notation allows to describe visually various data operations like projection, aggregation and join, plus ordered sequences of these operations. Additionally, the notation is flexible and can be easily extended with other operations on data if they would be helpful for the view modeler.

The advantages of our solution are: its simplicity and flexibility for the Grid designer; metamodel that supports semi-automatic generation of queries over and is well

suited for accompanying CASE modeling tools; the structure of meta-information database, which may be queried by DQP services and optimizers to compose views or move them in order to achieve additional query evaluation improvements. Grid users benefit from an abstract middle layer, which hides data integration and heterogeneity and is not limited to querying, but it supports all operations, which can be performed on virtual data shared in the Grid. The described views may be easily incorporated in OGSA-DAI based systems and similar solutions.

The future work will focus on visual modeling of scenarios that are more dynamic, and which could support usage of generic integration patters and transformation templates based on dynamical data ontology discovering.

References

1. S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. D. Ullman and J. Widom, The {TSIMMIS} Project: Integration of heterogeneous information sources. 16th Meeting of the Information Processing Society of Japan, Tokyo, 1994.
2. I. Foster, C. Kesselman, The Grid 2: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 2003
3. W. Heijenga. View definition in OODBS without queries: a concept to support schema-like views. In Doct. Cons. 2nd Intl. Baltic Wg on Databases and Information Sys-tems, Tallinn (Estonia), 1996
4. K. Kaczmarski, P. Habela, K. Subieta. Metadata in a Data Grid Construction. Proc. of the 13th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE-2004), Modena, Italy, 2004
5. H. Kozankiewicz, J. Leszczyłowski, K. Subieta. Updateable XML Views. Proc. of Advances in Databases and Information Systems (ADBIS), Springer LNCS 2798, pp. 385-399, Dresden, Germany, 2003.
6. M. Lenzerini. Data integration: a theoretical perspective. Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. Madison, Wisconsin. 2002
7. D. Calvanese, E. Damaggio, G. De Giacomo, M. Lenzerini, R. Rosati. Semantic Data integration in P2P systems. Proceedings of the International Workshop on Databases, Information Systems, and P2P Computing, Berlin, Germany, September 2003.
8. A.Y. Levy, A. Rajaraman and J. J. Ordille. Querying Heterogeneous Information Sources Using Source Descriptions. Proceedings of the 22nd VLDB Conference, Mumbai (Bombay), India, 1996
9. E. Naiburg, R. A. Maksimchuk. UML for Database Design. Addison-Wesley, 2001
10. Open Grid Services Architecture Data Access and Integration Documentation http://www.ogsadai.org.uk/dqp/
11. K. Subieta, C. Beeri, F. Matthes, and J. W. Schmidt. A Stack Based Approach to Query Languages. Proc. of 2nd Intl. East-West Database Workshop, Klagenfurt, Austria, September 1994, Springer Workshops in Computing, 1995.
12. I. Tatarinov, Z. Ives, J. Madhavan, A. Halevy, D. Suciu, N. Dalvi, X. Dong, Y. Kadiyaska, G. Miklau, and P. Mork. The Piazza Peer Data Management Project. ACM SIGMOD Record, 32(3), 2003.
13. T. Catarci, S. Chang, at al. A Graph-Based Framework for Multiparadigmatic Visual Access to Databases, IEEE Transactions on Knowledge and Data Engineering 8(3), 1996

Optimization of Distributed Queries in Grid Via Caching

Piotr Cybula[1], Hanna Kozankiewicz[2], Krzysztof Stencel[3], and Kazimierz Subieta[2,4]

[1] University of Lodz, Lodz, Poland
cybula@math.uni.lodz.pl
[2] Institute of Computer Sciences of the Polish Academy of Sciences, Warsaw, Poland
hanka@ipipan.waw.pl
[3] Institute of Informatics Warsaw University, Warsaw, Poland
stencel@mimuw.edu.pl
[4] Polish-Japanese Institute of Information Technology, Warsaw, Poland
subieta@pjwstk.edu.pl

Abstract. Caching can highly improve performance of query processing in distributed databases. In this paper we show how this technique can be used in grid architecture where data integration is implemented by means of updatable views. Views integrate data from heterogeneous sources and provide users with their integrated form. The whole process of integration is transparent, i.e. users need not be aware that data are not located at one place. In data grids caching can be used at different levels of architecture. We focus on caching at the middleware layer where the cache is stored in the database of the integrating unit. These results can be then used while answering queries from grid users, so there will be no need to reevaluate the whole queries. In such a way caching can highly increase performance of applications operating on grid. In the paper we also present an example how a query can be optimized by rewriting to make use of cached results.

1 Introduction

Performance is a crucial issue for distributed systems. In the area of databases research focuses on optimization of queries which allows for achieving better query evaluation time. There are many methods of distributed query optimization like query decomposition, using distributed indices, etc. Among these methods caching has emerged as one of fundamental techniques. Generally speaking caching increases performance via storing results of queries, which are then used during evaluation of following queries, so query processing is sped up. Cache can be kept at a middleware layer – i.e. a mediator between local servers (which provide data) and applications acting on these data.

Caching is a subject of various research papers like [1, 4, 6]. There exist many techniques related to caching e.g. cache investment [10]. In this method a query optimizer aims to enhance performance of future queries instead of thinking about performance of only currently processed query. It means that sometimes the query

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 387 – 396, 2005.
© Springer-Verlag Berlin Heidelberg 2005

optimizer generates a suboptimal plan for a particular query in order to establish a better performance of subsequent queries.

Data caching can be implemented by means of materialized views. There are a lot of work done in the area of processing queries using views e.g. [9]. One of the main problems is an algorithm determining choice of optimal views to materialize and there are many proposals [2, 8, 16]. There have been also proposed algorithms for answering queries using views that were developed for the context of data integration and distributed database systems [5, 15, 20]. In systems with data caching there is also a problem to have consistency of real data with cached data. The problem has been addressed in many papers, for instance [3, 7].

In this paper we aim to show how caching can be useful in our data grid architecture, where data are integrated by means of updatable object-oriented views with full computational power. Such views (defined in a very high-level query language) facilitate the development of intelligent mappings between heterogeneous data/service ontologies. Within the architecture we analyze how caching of data can be used to enhance performance. We show how cached results of queries stored at middleware layer can fasten query evaluation, when query is optimized via query rewriting techniques.

The paper is structured as follows. In Sections 2 and 3 we describe the Stack Based Approach and mechanism of updatable views defined within this approach. In Section 4 we present our grid architecture. Section 5 includes description how caching can be useful in such an application. In Section 6 we show an example of query optimization through rewriting that illustrates how cached results of queries can be used during query evaluation. Section 7 concludes.

2 Stack-Based Approach

Our grid mechanism is based on the Stack-Based Approach (*SBA*) to query languages [22, 23, 24]. *SBA* treats a query language as a kind of a programming language. Therefore, queries are evaluated using mechanisms which are common in programming languages. The approach is based on the *naming-scoping-binding* principle what means that each name in the query is bound to the proper run-time entity depending on the scope for the name. The mechanism of name binding and scopes are managed by means of environment stack (*ENVS*). The stack consists of sections and each of these sections describes a different environment e.g. environment of the databases, or of the user session. Sections of *ENVS* consist of entities called *binders* whose role is to relate a name with a run-time entity (object, procedure, view, etc). Binding of the name *n* means that the query interpreter looks through the consecutive sections of the *ENVS* to find the closest (to *ENVS* top) binder with the name *n*. The result of binding is a proper run-time entity (an object identifier, a physical memory address, etc.)

SBA defines an abstract formal framework that is known as the Stack-Based Query Language (*SBQL*). In *SBQL* queries can be defined in the following way:

- a literal or a name (of the variable, procedure, view, etc) is an atomic query
- more complex queries can be built from the simpler ones using unary operators (like *not*, *factorial, sin*) and binary operators (like **where**, *max*, +, ∀).

In *SBQL* all operators are orthogonal (with exception of typing constraints). In this way in *SBQL* can be assembled complex queries. Example *SBQL* queries are: *1*, *2+2*, *Book*, *Book* **where** *author* ="Lem", *Book*.(*title*, *author*).

SBQL also supports procedures and functional procedures with no restrictions on their computational complexity. Procedures can be defined with or without parameters, can have local environment, can call other procedures, and can have side-effects. Procedures are key elements of the *SBQL* view mechanism.

3 Updatable Views in SBA

A view is a mapping of stored data into virtual data. In the classical approach (SQL) a view is a function, similar to programming languages' functions. View updating means that a function result is treated as an l-value in updating statements. Such an approach, however, appeared to be inconsistent due to problems with finding unequivocal mapping of updates on virtual data into updates on stored data. In our approach to view updates [11, 12] a view definer can extend a view definition by the information on update intents. The information allows the programmer to eliminate ambiguities connected with multiple ways of performing a view update and the risk of warping user intention, which is the well-known problem related to view updates. In our approach a view definition consists of two main parts:

1. A mapping between stored and virtual objects. This part of the view definition has a form of a functional procedure that returns entities called *seeds*. They are used as parameters for the re-defining procedures. This part of the view definition is identified by the clause *virtual objects.*
2. Re-definitions of generic operations that can be performed on virtual objects. We have identified four such operations on virtual objects, i.e., *dereference* returning the value of a given virtual object, *insertion* of an object into a given virtual object, *deletion* of a given virtual object, and *update* the value of a given virtual object. The view definer has freedom to decide which of them and how these operations should be re-defined. (If an operation is not defined for virtual objects, it is forbidden). Description of these operations also has a form of functional procedures with arbitrary complexity. To distinguish operations we use fixed names *on_retrieve*, *on_insert*, *on_delete*, *on_update*.

   View definitions can contain other elements such as definition of subviews, internal state variables, etc.

View Update Process. A query interpreter must distinguish updates of virtual data and updates of stored data. Thus we have introduced the notion of a virtual identifier. It is a counterpart of an identifier of a stored object. A virtual identifier, besides information on the seed of a virtual object, contains information on the definition of view that has generated the given virtual object.

When the user performs an update operation on a virtual object, a query interpreter detects that it deals with a virtual object due to its virtual identifier. Thus instead of the generic update operation the interpreter calls the corresponding procedure defined within the view definition. The interpreter knows which operation is to be called due to view definition identifier included into the virtual identifier.

4 Grid Architecture

The heart of our approach is *the global virtual object and service store* (shortly: *global virtual store*), Fig.1 [13, 14]. *Global clients* are applications that send requests to the global virtual store. Global clients see data in grid according to *global schema* (collection of definitions of data/services provided by the global virtual store). The global schema is agreed upon by a consortium, by a standard or by a law that establishes the grid. The grid offers data/services supplied by *local servers*. *Local schemata* define data/services inside a local server. These schemata, as well as the business meaning of the data/services, can be different at each local server. The schemata are invisible to the grid users.

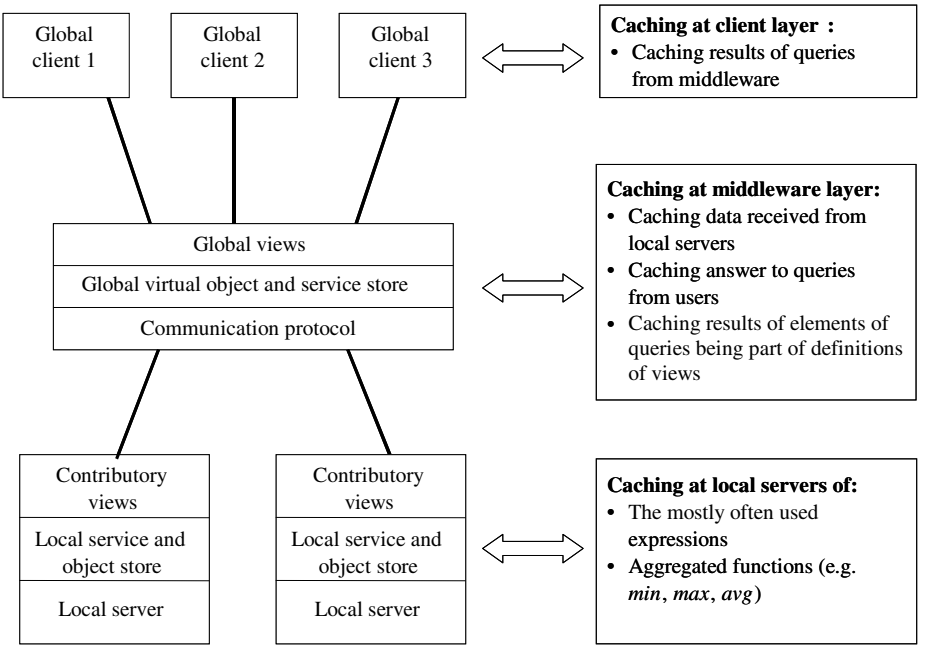


Fig.1. Architecture of the GRID illustrating places where caching can be used

The first step of integration of a local server into the grid is done by the administrator of this local server who has to define the *contributory schema*. It is the description of the data and services contributed by the local server to the grid. The local server's administrator also defines *contributory views* that constitute the mapping of the local data/services to the canonical data/service model assumed by the grid. The second step of the integration of local servers into the grid is the creation of *global views*. These views are stored inside the global virtual store. The interface of them is defined by the global schema. They map the data and services provided by the local servers to the data and services available to the global clients.

The global views are defined by the *grid designer*, which is a person, a team or software that generates these views upon the contributory schemata, the global

schema and the integration schema. The *integration schema* contains additional information how particular data and services of local servers contribute to the global canonical model of the grid. In Fig.1 we show that caching can be used at each of three layers of the architecture: at clients, at middleware, and at local servers.

Data Integration Example. Assume a database that contains *Book* objects with attributes *ISBN*, *title*, *author* and *price*. In this paper we based our consideration on an example of integration where data are horizontally fragmented (it is one of the most frequent case). Assume there are three local servers located in Cracow, Gdansk, and Warsaw (we call these servers by their locations). Each server keeps data of the books. All contributory schemata are the same. In the considered case the integration schema contains the following information:

- ISBN is a unique identifier of all books.
- There are no duplicates of *ISBN* in local databases.
- The global database is the virtual union of all three databases.

We would like to define a view that provide prices and titles of books with prices smaller than 5 Euro. The definition of the global view might look as shown below:

```
create view CheapBookDef {
   virtual objects CheapBook {    /* returning seeds */
     return ((Gdansk.Book ∪Cracow.Book∪Warsaw.Book) where price < 5) as b;}
   on_delete do { delete b }

   create view TitleDef {
      virtual objects Title { return b . title as t; }
      on_update newTitle do { t := newTitle; }
      on_retrieve do { return t; }
    }

    create view PriceDef {
       virtual objects Price { return b . price as p; }
       on_retrieve do { return p; }
     }
}
```

Note that the view refers to objects from a particular server using the name of its location (e.g. **delete** *Warsaw . Book*). The view delegates the updates to appropriate servers. The view definer can use implicitly several routines of the communication protocol, for instance, the navigation ("**.**"), **insert**, **delete**, update (":="). These remote operations are called as if they were local.

We would like to emphasize that the global clients of the view cannot see the origin of particular virtual objects. The seeds of virtual objects are encapsulated and global clients just refer to the name of the view and attribute names, like in the following query that returns the titles of all very cheap books:

(*CheapBook* **where** *Price* < 2) . *Title*

or in an updating statement (change the title 'Salaris' into 'Solaris'):

(*CheapBook* **where** *Title* = 'Salaris') . *Title* := 'Solaris';

5 Caching in Grid Architecture

In our architecture caching of data can be used in threefold (refer to Fig.1):

1. *Caching at local servers*. This kind of caching is identical to caching in non-distributed databases. The problem has been described in details in [3].
2. *Caching at clients.* Applications acting on grid can store results of queries returned by grid and use this results later instead of re-asking grid database. It seems that this kind of caching does not imply any conceptual difficulties.
3. *Caching at middleware*. Caching at integrating unit can have many forms:
   - Caching of data stored at local servers (e.g. if server *A* stores data of books, then this data are kept replicated in grid data store).
   - Caching results of queries that are often evaluated while creating virtual objects (e.g. if average price of books at servers *A*, *B*, and *C* is often evaluated then we can save this value in the object store).
   - Caching answers to client queries – in this way query reevaluation can be avoided.

In this paper we focus on the last category of caching i.e. data caching at middleware. The choice of queries beneficial for caching is an important issue and there are plenty of techniques which allows choosing a set of optimal queries for caching. Majority of such techniques is based on statistics kept at clients (how often parts of queries are used, mean access time to servers, etc).

We assume that at middleware there must be a cache registry – a module which keeps information about all stored results of queries. The organization of this cache has been described in [3].

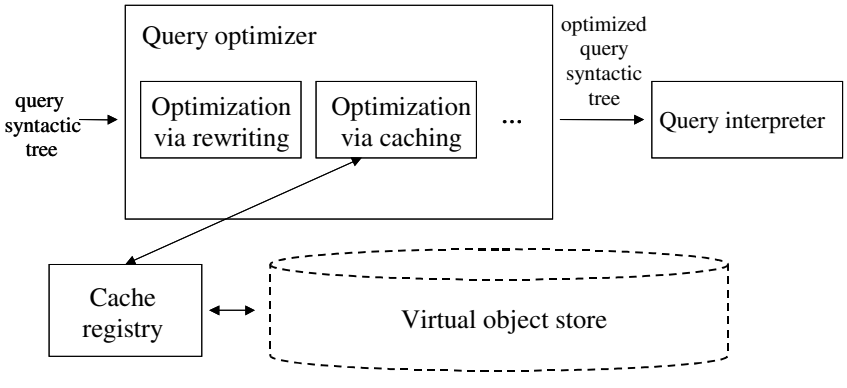


Fig.2. Architecture of query evaluation environment at middleware

In Fig.2 we presented architecture of query optimization at middleware layer. (We included only techniques related to caching.) The scenario of query processing looks as follows – the middleware receives query from a user. The parser transforms it into the form of a syntactic tree. This tree is the input of the middleware query optimizer which rewrites it using known rewriting rules. The optimizer uses the cache registry to identify queries that need not be reevaluated. At the end the optimizer passes the optimized query syntactic tree to the query interpreter.

6 Query Optimization Via Rewriting

Due to full orthogonality and consistent conceptual continuation (with no anomalies and irregularities), queries involving views are fully optimizable through the query modification technique [21, 25]. The idea of query modification is to combine a query containing a view invocation with the definition of the invoked view. The resulting query has no references to view invocations (instead, it has direct access to their definitions). The optimization concerns cases when sack definitions are reduced to single (however complex) queries, with no parameters and recursion. These conditions are satisfied by majority of views. In all such cases textual substitution of the views invocation by the corresponding query from the sack definition results in a semantically equivalent query that can be optimized by standard methods, e.g. by removing dead sub-queries, factoring out independent sub-queries, and low level techniques (e.g. based on indices).

Some peculiarity of our view mechanism in comparison to [25] concerns the fact that a view for retrieval is determined by two queries: the first one from the sack definition and the second one from the *on_retrieve* clause. In consequence, we proceed as follows: the first query substitutes textually the corresponding view invocation, and the second query substitutes textually the dereferencing operator acting on the view invocation (in the post-fix order, with the dot operator inside). Although the dereferencing operator is usually implicit in queries, it can be easily recognized during parsing. Thus, we can apply the full static optimization by query modification.

Updating operations that are parts of a view definition do not interfere with the use of the presented methods, but only for querying. However, view updating requests cannot be optimized in this way. New optimization techniques addressing this issue will be the subject of further investigations.

After query modification several rewriting techniques can be used to achieve an optimized form of the query, among others:

- Removing dead subqueries, i.e., these parts of the query which do not contribute to the result of the whole query in any way [19].
- Method of independent subqueries [18] that is based upon the observation that some subqueries can be evaluated outside loops implied by non-algebraic operators.
- Changing joins into less expensive navigations.
- Removing auxiliary names.
- Transforming queries to forms where indices can be used.
- Using cached results of queries.

Example of Query Optimization. To illustrate query rewriting technique we analyze a query that involves a view *CheapBook* defined in Section 4. The view returns virtual objects that include information about titles and prices of books that are claimed to be cheap (i.e. they cost less than 5 Euro).

Here, we present the consecutive steps of optimization of a query that contains a view invocation. We assume that server in Cracow has very long access time and therefore data of books kept there are cached at middleware (under the name *CBook*). Let us consider the following query, that returns titles of cheap books, which price is lower then the average price of cheap books with title “databases”:

(*CheapBook* **where**
Price < *avg*((*CheapBook* **where** *Title* = “databases“) . *Price*)
) . *Title*

The optimization of the query may look as follows:

1. In the first step we add explicit calls to the dereference function in these places where the dereference is implicit (e.g. it is forced by operator =):

(*CheapBook* **where** *deref* (*Price*) <
avg((*CheapBook* **where** *deref* (*Price*) = “databases“) . *deref* (*Price*))
) . *deref* (*Title*)

2a. In the next step we substitute the view invocation with parts of its definition. For the sake of clarity we divided this step into two (2a and 2b). First, we insert a query from *on_retrieve* body of the corresponding view instead of calls to the *deref* function:

(*CheapBook* **where** (*Price* . *p*) <
avg(((*CheapBook* **where** (*Title* . *t*) = “databases“) . (*Price* . *p*)))
) . (*Title* . *t*)

2b. Next, we substitute all view invocations by the queries from the corresponding sack definitions:

((((*Gdansk.Book* ∪*Cracow.Book*∪*Warsaw.Book*) **where** *price* < 5) **as** *b*)
where ((*b* . *price* **as** *p*) . *p*) <
avg(((((*Gdansk.Book* ∪*Cracow.Book*∪*Warsaw.Book*) **where** *price* < *5*) **as** *b*)
where ((*b* . *title* **as** *t*) . *t*) = “databases“) . ((*b* . *price* **as** *p*) . *p*))
) . ((*b* .title **as** *t*) . *t*)

3. In the following step we remove auxiliary names *t* and *p*:

((((*Gdansk.Book* ∪*Cracow.Book*∪*Warsaw.Book*) **where** *price* < 5) **as** *b*)
where (*b* . *price*) <
avg(((((*Gdansk.Book* ∪*Cracow.Book*∪*Warsaw.Book*) **where** *price* < 5) **as** *b*)
where (*b* .*title*) = “databases“) . (*b* . *price*))
) . (*b* . *title*)

4. Next, we remove the auxiliary name *b*:

(((*Gdansk.Book* ∪*Cracow.Book*∪*Warsaw.Book*) **where** *price* < 5)
where *price* <
avg((((*Gdansk.Book* ∪*Cracow.Book*∪*Warsaw.Book*) **where** *price* < 5)
where *title* = “databases“) . *price*)
) . *title*

5. We connect two consecutive *where* subqueries into one subquery with the filtering condition built by means of the *and* operator:

((*Gdansk.Book* ∪*Cracow.Book*∪*Warsaw.Book*) **where** *price* < 5
and *price* < *avg*(((*Gdansk.Book* ∪*Cracow.Book*∪*Warsaw.Book*)
where *price* < 5 **and** *title* = “databases“) . *price*)
) . *title*

6. Now, we take the common part before a loop to prevent multiple evaluation of a query that calculates average price of cheap books on databases:

(*avg*(((*Gdansk.Book* ∪*Cracow.Book*∪*Warsaw.Book*) **where** *price* < 5
and *title* = "databases") . *price*) **as** *a*)
.((*Gdansk.Book* ∪*Cracow.Book*∪*Warsaw.Book*) **where** *price* < 5 **and** *price*<*a*) . *title*

7. Finally, we use the cached result *CBook* (because the access to the data from Cracow is very slow):

(*avg*(((*Gdansk* . *Book* ∪ *CBook* ∪ *Warsaw* . *Book*) **where** *price* < 5
and title = "databases") . *price*) **as** *a*)
. ((*Gdansk* .*Book* ∪ *CBook* ∪*Warsaw* . *Book*)**where** *price* < *5* **and** *price* < *a*) . *title*

The above query is optimal according to the assumed rewriting methods. We would like to underline that all steps of query optimization presented in this section can be performed in an automatic way using query optimization techniques elaborated for *SBQL* ([17]).

7 Conclusions and Future Work

In the paper we have described how query caching can be used to enhance performance of applications operating on grids. We based our consideration on grid architecture where the key element is the mechanism of updatable views. We showed where in this architecture caching can be implemented and how query rewriting techniques can use these cached results.

Future research on caching in grid will include:

- Elaboration of methods of selection of optimal query set for caching.
- Providing mechanism of consistency of data in cache at middleware and data stored at local servers (protocol is required that will propagate such updates).
- Work on new method of query rewriting to be able to optimize queries involving complex view definitions (current methods work on views, where the sack definition and operations are limited to a single query).

In the future we are aiming to implement caching in distributed grid database based on the ODRA prototype – an object-oriented database platform built from scratch by our team.

References

1. S. Adali, K.S. Candan, Y. Papakonstantinou, V.S. Subrahmanian: Query Caching and Optimization in Distributed Mediator Systems. SIGMOD Conference 1996: 137-148
2. S. Agrawal, S. Chaudhuri, V.R. Narasayya: Automated Selection of Materialized Views and Indexes in SQL Databases. VLDB 2000: 496-505
3. P. Cybula, K. Subieta: Cached Queries in the Stack Based Approach. ICS PAS Report 985, 2005

4. S. Dar, M.J. Franklin, B.T. Jónsson, D. Srivastava, M. Tan: Semantic Data Caching and Replacement. VLDB 1996: 330-341
5. O.M. Duschka, M. Genesereth, A.Y. Levy: Recursive Query Plans for Data Integration. Journal of Logic Programming. Logic Based Heterogeneous Information Systems, 2000
6. M..J. Franklin, M.J. Carey, M. Livny: Local Disk Caching for Client-Server Database Systems. VLDB 1993: 641-655
7. M..J. Franklin, M.J. Carey, M. Livny: Transactional Client-Server Cache Consistency: Alternatives and Performance. ACM Trans. Database Syst. 22(3): 315-363 (1997)
8. H. Gupta, I.S. Mumick: Selection of Views to Materialize in a Data Warehouse. IEEE Trans. Knowl. Data Eng. 17(1): 24-43 (2005)
9. A.Y. Halevy: Answering queries using views: A survey. VLDB J. 10(4): 270-294, 2001
10. D. Kossmann, M.J. Franklin, G. Drasch: Cache investment: integrating query optimization and distributed data placement. ACM Trans. Database Syst. 25(4): 517-558 (2000)
11. H. Kozankiewicz, J. Leszczyłowski, K. Subieta: Updatable XML Views. Proc. of ADBIS Conf., Springer LNCS 2798, pp. 385-399, Dresden, Germany, 2003
12. H. Kozankiewicz, J. Leszczyłowski, K. Subieta: Implementing Mediators through Virtual Updatable Views. Proc. of EFIS Workshop, IOS Press, pp. 52-62, Coventry, UK, 2003
13. H. Kozankiewicz, K. Stencel, K. Subieta: Integration of Heterogeneous Resources through Updatable Views. ETNGRID Workshop, June 2004, Proc. published by IEEE
14. H. Kozankiewicz, K. Stencel, K. Subieta: Implementation of Federated Databases through Updatable Views. Proc. of the European Grid Conference, Amsterdam, The Netherlands, 2005, LNCS (to appear)
15. A.Y. Levy, A. Rajaraman, J.J. Ordille: Querying Heterogeneous Information Sources Using Source Descriptions. VLDB 1996: 251-262
16. H. Mistry, P. Roy, S. Sudarshan, K. Ramamritham: Materialized View Selection and Maintenance Using Multi-Query Optimization. SIGMOD Conference 2001
17. J. Płodzień: Optimization Methods in Object Query Languages. Ph.D. Thesis. Institute of Computer Science, Polish Academy of Sciences (2000)
18. J. Płodzień, A. Kraken: Object Query Optimization through Detecting Independent Subqueries. Inf. Syst. 25(8): 467-490 (2000)
19. J. Płodzień, K. Subieta: Query Optimization through Removing Dead Subqueries, Proc. of ADBIS Conf., Springer LNCS 2151, 2001, 27-40
20. R. Pottinger, A.Y. Halevy: Minicon: A Scalable Algorithm for Answering Queries Using Views. VLDB Journal 2001
21. M. Stonebraker: Implementation of Integrity Constraints and Views by Query Modification. Proc. of SIGMOD Conf., 1975
22. K. Subieta: Theory and Construction of Object-Oriented Query Languages. Polish-Japanese Institute of Information Technology Editors, Warsaw 2004, 522 pages
23. K. Subieta, C. Beeri, F. Matthes, J.W. Schmidt: A Stack Based Approach to Query Languages. Proc. of Intl. East-West Database Workshop, 1995
24. K. Subieta, Y. Kambayashi, J. Leszczyłowski: Procedures in Object-Oriented Query Languages. Proc. of 21-st VLDB Conf., 1995, 182-193
25. K. Subieta, J. Płodzien: Object Views and Query Modification, Proc. of IEEE BalticDB&IS, pp. 13-24, Vilnius, Lithuania, 2000

Modelling the √N + ROWA Model Approach Inside the WS-ReplicationResource

Manuel Salvadores[1], Pilar Herrero[2], María S. Pérez[2], and Alberto Sanchez[2]

[1] IMCS, Imbert Management Consulting Solutions,
C/ Fray Juan Gil 7, 28002 Madrid , Spain
mso@imcs.es
[2] Facultad de Informática, Universidad Politécnica de Madrid,
Campus de Montegancedo S/N, 28.660 Boadilla del Monte, Madrid, Spain
{pherrero, mperez, ascampos}@fi.upm.es

Abstract. This paper presents the √N + ROWA model that is been develop at the Universidad Politécnica de Madrid with the aim of replicating information in Grid environments and optimizing the number of messages to be exchanged in the process as well as their use to built Grid environments based on WSRF specifications. The model presented in this paper will be one of the pillars of a new Grid service (WS-ReplicationResource) that, in the near future, will provide Grid systems with a high level service of resource replication.

1 Introduction

The last tendencies of WSRF (Web Services Resource Framework) specifications oriented towards Grid Computing Systems development and their implementation in Globus Toolkit 4 (GT4) [13] will mark, in the near future, the main development lines around middleware to support Grid applications based on the OGSA standard [1].

OGSA enumerate those characteristics that Grid systems have to possess. High availability[1] plays an important role among all these characteristics. The replication concept is close related to the availability concept, being one of the techniques more employed for failure recovery.

The WS-ResourceProperties specification [2], as a part of WSRF specifications, defines an standard way to exchange messages that could allow a client to consult or update the values of the properties associated to each specific resource.

A resource could be defined as the Web Service (WS) that having a set of properties, defined by the WS-ResourceProperties, and being its state the combination of all the values associated to all these properties at a given moment, can maintain this state through the WS-Addressing [14].

To have the information related to each of these specific resources replicated will be quite useful not just to allow a high availability in the system but also to design new collaborative models as well as to introduce complex negotiation models and mechanisms based on agents.

On the other hand, this model of synchronisation could be extended to high-scale transactional systems as component integrated inside a framework, such as DCP-Grid [11, 12].

[1] OGSA specification, point 2.10.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 397 – 405, 2005.
© Springer-Verlag Berlin Heidelberg 2005

In this paper we present our specification, the one we have called WS-ReplicationResource, extending all the functionalities of WS-ResourceProperties to allow the replication of the properties of a WS through the nodes connected to a Grid infrastructure.

This paper has been organized as follows. The next section exposes the motivation of this work. The paper continues with a brief discussion about the related work on the area, our approach and the scalability of the system under our approach. The paper concludes with a section to present the paper's conclusions and the ongoing and future work.

2 Motivation

Four operations are defined in the WS-ResourceProperties specification to access to the resource's properties:

1. GetResourceProperty: to obtain the property's value.
2. GetMultipleResourceProperty: to obtain the value associated to several properties in just one operation.
3. SetResourceProperties: to create update properties in just one resource.
4. QueryResourceProperties: to carry out some queries, related to a specific resource's properties, through XPath [15].

Taking into account all these operations, our motivation is to propose a decentralised scenario in which each of the nodes could be the queries' or updates receptor over an specific replicated resource, having an idea about which of the resource's properties could be accessed at a given moment and making all this possible in an autonomous way.

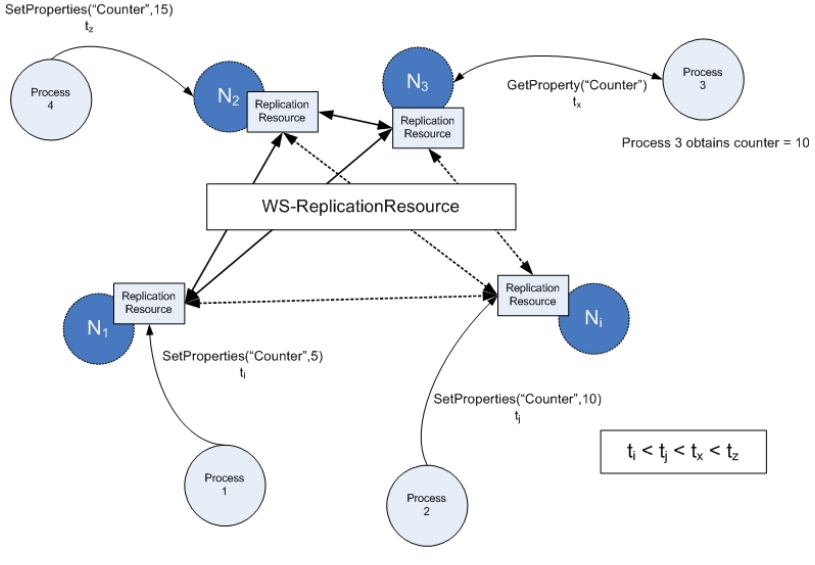


Fig. 1. Scenario: Four actions are brought about a replicated resource, WS-ReplicationResource must ensure the causal order

Figure 1 presents the initial scenario to be solved taking into account i nodes $N=\{N_1,N_2,N_3, \ldots, N_i\}$. Each of these nodes could also be the receptor of reading requests as well as writing. In order to ensure the casual order (fairness)[10] of the actions to be carried out, we could represent each of the actions as a tuple (a, t), where 'a' represents the action to be carried out in the moment 't'. In this way, if A is a sequences of 4 actions (N=4), A could be represented as:

$$A=\{ (a1,t1), (a2,t2), (a3,t3) ,(a4,t4) \} \quad (1)$$

The casual order would imply to introduce new constraints such as:

$$\forall i = 1, 2, 3...N \; t_i < t_{i+1} \rightarrow A_i \; before \; A_{i+1} \quad (2)$$

In the figure 1 it is possible to appreciate the problematic situation caused by action 3 due to its counter's value which should be 10. The casual order in the execution of reading/writing operations could be solved by a controlled access to the resources in mutual exclusion.

3 Related Work

One of the first algorithms used for access in a mutual exclusion is the Ricart y Agrawala algorithm [3]. Ricart and Agrawala's Algorithm solves the synchronization problem in distributed systems. This algorithm insures that only one process will be allowed in a critical region at a time. It works by using a system of messages and acknowledgements. The sending of a message is assumed to be reliable; that is, every message is acknowledged. The algorithm works as follows:

When a process wants to enter a critical region, it builds a message containing the name of the critical region it wants to enter, its process number and the current time. It then sends the message to all the other processes including itself. When a process receives a request message from another process, the action it takes depends on its state with respect to the critical region named in the message. There are three possible states:

- If the receiver is not in the critical region and does not want to enter it, it sends back an OK message to the sender (shown as Ready state in workbench).
- If the receiver is already in the critical region, it does not reply. Instead, it queues the request (shown as In CS state in workbench).
- If the receiver wants to enter the critical region, but has not yet done so, it compares the timestamp in the incoming message with the one contained in the message that it has sent everyone. The lowest one wins. If the incoming message is lower, the receiver sends back an OK message. If its own message has a lower timestamp, the receiver queues the incoming message and sends nothing (shown as waiting state in workbench).

After sending out requests asking permission to enter a critical region, a process sits back and waits until everyone else has given permission. As soon as all the permissions are in, it may enter the critical region. When it exits the critical region, it sends OK messages to all processes on its queue and deletes them all from the queue.

This algorithm will grow up proportionally to the number of nodes needed because it would be necessary: 2*(N-1) messages, being N the number of nodes, to became to an agreement in the critical section; (N-1) messages to let the rest of the nodes know that I would like to access to the critical section; and (N-1) answers from he rest of the nodes to give the final approval.

The algorithms based on Quorums [8], are the optimised option to access to critical sections in distributed systems, where Quorum could be defined as:

> *"Let S = {S1, S2, …} be a set of sites. A quorum system Q is a set of subsets of S with pair-wise non-null intersection. Each element of Q is called a quorum".*

For example, if we have four sites, S1, S2, S3 and S4. A possible quorum system then consists of these three quorums: {S1, S2, S3}, {S2, S3, S4} and {S1, S4}, although there are many other possible quorum systems for these four sites.

In systems based on Quorums a process can access to the critical section if an only if it obtains the permission of all the elements of its own Quorum. This is a way of reducing, considerably, the number of messages.

Quorums could be combined with the ROWA technique [5] (Read One-Write All). This technique introduces a difference in between writing and reading operations as follow:

- Read Operations: read from any site. If a site is down, try another site.
- Write operations: write to all sites. If any site rejects the write, abort the transaction.

But the ROWA technique is not working if one of the repliques fails, and therefore the combination with Quorums is one of the techniques more useful.

As for the combination Quorums + ROWA [6], it is important to highlight that in this case there are two different kinds of Quorums: writing (wq) and reading (rq). Both of them will have the following constraints in the Read-One-Write-All (ROWA) approach:

- Logical read on data item x is converted to physical read on any of its copies ("read one")
- Logical write of x is translated to physical write on all of the copies of x.

On the other hand, the different topologies and negotiation policies inside the Quorums allow to distinguish among numerous types of different Quorums, being some of them:

- Majority o Consensus [8]: Uses voting to reach consensus. Each site has an assigned weight (number of votes), and quorums are defined so that number of needed votes exceeds half of the total (majority). If n is the sum of all assigned weights, then read (rq) and write (wq) quorums must then fulfill $2 * |wq| > n$ and $|rq| + |wq| > n$ and the Minimum quorum sizes that work will be $|wq|=n/2+1$ $|rq|=n/2$.
- Tree: A generalization of Majority Quorum. The main idea is to organize the sites into a hierarchy. This hierarchy is represented as a complete tree where physical sites appear at the leaves of the tree. At each level (starting at the root level) of the tree, a majority of tree nodes must be chosen. For each node chosen at *level i,* a majority of nodes at *level i+1* must be chosen. A *write*

quorum consists of the root of the tree, a majority of its children, a majority of the children of each children, etc. A *read quorum* consists of the root of the tree. If the root is unavailable, the read quorum consists of a majority of its children, and so recursively [9].

- Grid: There are different kinds of Grid quorum, such as the rectangular or the triangular. In the rectangular, a read quorum consists of an element of each column (|rq| = c) A write quorum requires an entire column and one element from each of the remaining columns (|wq| = r + c - 1) If the grid is a square |rq| = vn |wq| = 2 * vn -1 [7].

Finally, the √N algorithm, being N the number of nodes, is based on the association of nodes in N minimum subsets with no null intersection (between each two of them) [4]:

$$\forall i, j\, 1 \leq i, j \leq N, S_i \cap S_j \neq \varnothing \tag{3}$$

Taking into account the different methods and algorithms presented here, in the next section we will introduce our approach which is based on the association of two of these algorithms: √N algorithm [4] and the ROWA technique [5].

4 The √N + ROWA Model

Initially we will identify two components to be introduced inside each and every node (see figure 2):

1. A mutex property deployed in the nodes as a WS-ResourceProperty.
2. The √N + ROWA engine that interacts with the mutex to take decisions and implements the read and write operations.

As will be possible to appreciate along this section, these components could be considered key factors achieve replication Once a resource has the mutex property it can subscribe, through the WS-Notification [17], to others services (or nodes) that belongs to its group (quorum). Due to the subscriptions between quorums the nodes can be informed about the state of other mutexs.

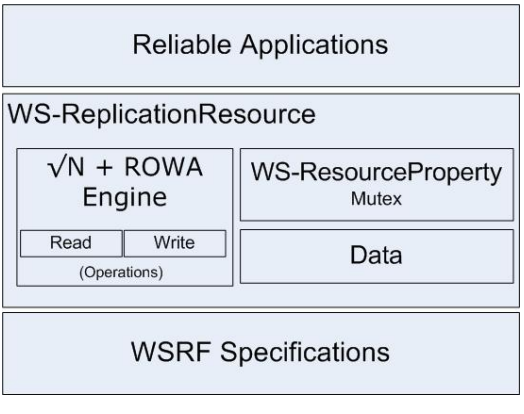


Fig. 2. The √N + ROWA model architecture

The √N+ ROWA model, takes into account the √N protocol [4] and applying the ROWA protocol to control the access based on the different types of operations (writing and reading). It makes also distinguish in between the reading and the writing quorums (rq and wq respectively). On the other hand, in order to replicate the data through the nodes, our approach will consider two key factors:

a) The impact that the "*lazy propagation*" technique will have over the model
b) The scalability of the system.

In our model, when an i-node wish to carry out an writing operation, it requires the votes of the quorum S_i and the writing information will be replicated only in those elements of its quorum. In order to carry out a writing/reading operation over S_j being i≠j and $S_i \bigcap S_j = N_z$, the node N_z will have to send S_j the updated modifications over the synchronised element before giving S_j its vote.

In the figure 3, it is possible to appreciate the "Lazy Propagation" effect because the operation 1 (write request over the node 2), which requires the obtaining of wq, repliques the writing operation only to the rest of the S_2 nodes (operation 2). In this moment *t_counter* is increased in the S_2 nodes. When the node 4 receives a read request (operation 3), and while this node is negotiating this request, the nodes 4 and 2 detect $t_counter_2 > t_counter_4$ and therefore they would need to update their values (operation 4). Something similar would happen if the node 1 receives a writing request, operation 6.

Taking into account all this possible situations, we could define that a system is replicabily stable if:

$$\forall i, j\, 1 \le i, j \le N \rightarrow t_counter_i = t_counter_j \tag{4}$$

Although the system stability is an important issue in this kind of systems, another important factor is to keep this system stability while the system scalability increases.

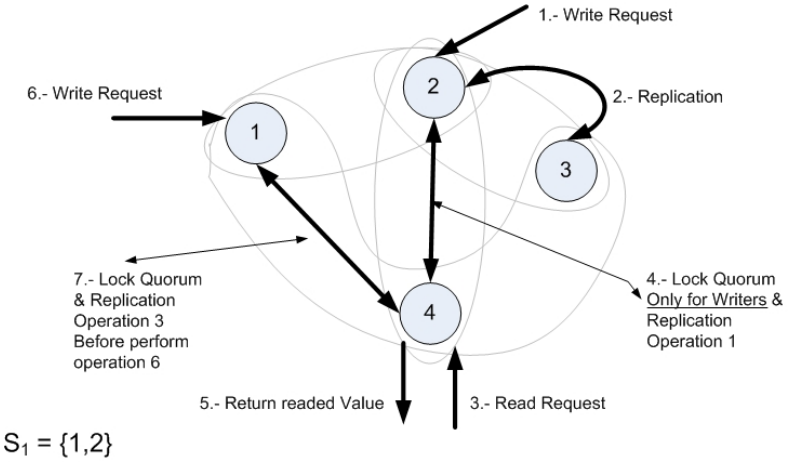


Fig. 3. √N+ ROWA model interaction based on S1...S4 Quorums and 4 nodes

In the next section, we will present a study that we have performed with the aim of modelling not just reading/writing operations to be carried out in the system, but also the cost of the replication model to became as stable as we have defined in this paper by the formulas 3 and 4.

5 Scalability

Our first work hypothesis, in this study, will be that there are not possible collisions in the system, leaving this case of study as future research work, and the second one will be that the time to process one operation is much lower that the number of transactions per unit of time. That is:

$$t_{proc} <<< {}^{n_pet}/_{sg.} \rightarrow p(colision) \simeq 0 \tag{5}$$

This implies that the system has time enough to recover from a lock for accessing to a resource before receiving another request.

The average of messages to be sent would be the addition of (k-1) from the operation request, (k-1) answers, (k-1) to the replication (only in a writing request) and (k-1) only if the last operation was carried out in another Quorum (see equation 6). To see equation 6 demonstrations go to [4].

$$m = (k-1) + (k-1) + p(w)(k-1) + p(change_q)\,p(w)(k-1) \tag{6}$$

Being *p(change_q)* the probability that two sequential operations are executed in different quorums, *p(w)* the probability that that the operation is writing and k the quorum length. Moreover, if N is the number of nodes and *k* is the size of the

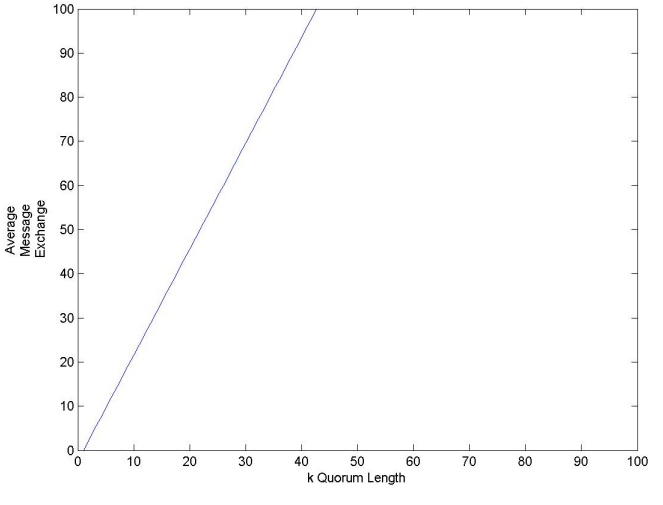


Fig. 4. Average message exchange applying a lazy propagation for those Quorums which length is 0..100

intersection set no linear between each of them, then the equation 7 will complement to the previous one:

$$N = k^2 - k + 1 \text{ [4]} \tag{7}$$

On the other hand, if want that the computational process will take advantage of the balance capability, the probability of Quorum's change *p(change_q)* should follow the equation 8:

$$p(change_q) = N - 1 / N \tag{8}$$

Equations 6, 7 and 8 will allow us to obtain a the average message exchange as a function of the quorum length and, if the likelihood for a writing request is p(w) = 0.2, we could represent this function as it is showed in figure 4. The average message exchange for different quorums' length is showed in table 1.

Table 1. Average message exchange for different quorums' length

Number Nodes	Quorum Length (K)	Messages Exchange
381	20	42
1561	40	86
3541	60	130
6321	80	174

In the table 1 it is possible to appreciate the system scalability. When the number of nodes is close to 381, the average of messages to be exchanged to access to the exclusion mutual area is close to 42, being this value quite acceptable. However, if the number of nodes increases to 1561 (four times), the number of messages to be exchanged will be double. As it is possible to appreciate in the table 1 and figure 4, the number of messages to be exchanged is proportional to the Quorums' length (K). However, the quorums' length will be almost the square root of the number of nodes (scalability) (see equation 7).

6 Conclusions and Future Work

This paper describes a model that is been carried out at the Universidad Politécnica de Madrid with the aim of replicating the information optimizing the number of messages to be exchanged as well as their use to built Grid environments based on WSRF specifications. The approach presented in this paper will be one of the pillars of a new grid service: WS-ReplicationResource.

As ongoing work, we are currently working on its implementation and in a near future we are planning the deployment in a large scale grid infrastructure, providing high level of transancionality of actions to be carried out inside the environment.

References

1. Foster, I., Kesselman, C., Nick, J., Tuecke, S. *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*. Open Grid Service Infrastructure WG, Global Grid Forum, 2002.
2. Graham S., Treadwell J. *Web Services Resource Properties 1.2 (WS-ResourceProperties)*, Working Draft 04. http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-04.pdf, 2004.
3. Ricart G., Agrawala A.K *An optimal algorithm for mutual exclusion in computer networks*. Communications of the ACM 24, pp. 627-628, 1981.
4. Maekawa M.: *A Square Root N Algorithm for Mutual Exclusion in Decentralized Systems.* ACM Transactions Computer Systems 3(2), pp. 145-159, 1985.
5. Jiménez-Peris R., Patiño-Martínez M., Alonso G., Kemme B.. *How to Select a Replication Protocol According to Scalability, Availability, and Communication Overhead*. Proceedings of the 20th IEEE International Conference on Reliable Distributed Systems, SRDS'01, pp. 24-33, New Orleans, 2001.
6. Bernstein P. A., Hadzilacos V., and Goodman N. *Concurrency Control and Recovery in Database Systems*. Addison Wesley, Reading, MA, 1987.
7. Cheung S. Y., Ammar M. H., Ahamad M.: *The Grid Protocol: A High Performance Scheme for Maintaining Replicated Data*. IEEE Transactions on Knowledge and Data Engineering. 4(6), pp. 582-592, 1992.
8. Thomas R. H. *A Majority Consensus Approach to Concurrency Control for Multiple Copy Databases*. ACM Transactions on Database Systems, 4(9), pp. 180–209, 1979.
9. Agrawal D. and Abbadi A. E. *The Tree Quorum Protocol: An Efficient Approach for Managing Replicated Data*. In Proceedings of the 16th VLDB Conf., Brisbane, Australia, 1990.
10. Kenneth P. Birman: *Building Secure and Reliable Network Applications*. In proceeding of the Worldwide Computing and Its Applications, International Conference (WWCA '97), pp. 15-28, Tsukuba, (Japan), 1997.
11. Salvadores M., Herrero P., Pérez María S., Robles V.*DCP-Grid, A Framework for Concurrent Distributed Transactions on Grid Environment*. The First International Workshop on "Knowledge and Data Mining Grid" (KDMG'05) On the 3RD Atlantic Web Intelligence Conference 2005 (AWIC'05), LNAI 3528- 0498, Lodz (Polonia) 2005.
12. Salvadores M., Herrero P., Pérez María S., Robles V. DCP-Grid, *A Framework for Conversational Distributed Transactions on Grid Environment*. International Workshop on Grid Computing Security and Resource Management (GSRM'05) In conjunction with the International Conference on Computational Science 2005(ICCS 2005), LNCS 3516, Emory University Atlanta (USA), 2005
13. Globus Toolkit: http://www.globus.org/toolkit/ Consulted in June 2005.
14. WS-Addressing, Worl Wide Web Consortium: http://www.w3.org/Submission/ws-addressing/ Consulted in June 2005.
15. XPath, World Wide Consortium,: http://www.w3.org/TR/xpath Consulted in June 2005.
16. Kenneth P. Birman: *The Process Group Approach to Reliable Distributed Computing. Commun*. ACM 36(12), pp. 36-53, 103 1993.
17. WS-Notification, IMB Developer Works http://www-106.ibm.com/developerworks/library/specification/ws-notification/ Consulted in June 2005.

MIOS+INTEROP 2005 PC Co-chairs' Message

Modern enterprises face a strong economical pressure to increase competitiveness, to operate on a global market, and to engage in alliances of several kinds. Agility has become the new guiding principle for enterprises. It means that enterprises must be able to easily collaborate with other enterprises expand, be it through participating in enterprise networks, or through mergers or acquisitions, or through insourcing or outsourcing of services. In order to meet these economical requirements, enterprises rely increasingly on the benefits of modern information and communication technology (ICT). However, the appropriate knowledge to deploy this technology as needed, and in an effective and efficient way, is largely lacking, particularly knowledge regarding the collaboration of enterprises and knowledge regarding the interoperability of their information systems.

Successful collaboration between enterprises requires flexible organizational structures and business processes, which must be based on shared business process models and application semantics, in order to produce goods and services quickly and at low cost while maintaining high quality. The successful interoperability of their information systems requires in addition flexible supporting ICT-applications and flexible ICT-infrastructures, which must be based on shared data exchange formats and communication protocols.

INTEROP is a Network of Excellence that facilitates the emergence of interoperability research issues through the fusion of three domains, namely: architectures, enterprise modeling, and ontologies. The architectural perspective is focusing on implementation frameworks for interoperability, the enterprise modeling perspective is defining interoperability requirements, and the ontological perspective is dealing with semantics issues for interoperability in the enterprise. The MIOS-INTEROP workshop is a concise reference to the state of the art in *Modeling Inter-Organizational Systems* and *Interoperability of Enterprise Software and Applications* that will be of great value to engineers and computer scientists working in manufacturing and other process industries and to researchers coming from the academic environment.

Composed of 18 selected papers, out of 43 submitted, of international authorship, the workshop ranges from academic research through case studies to industrial experience of interoperability and inter-organisational systems modeling. A lot of pioneering work still needs to be done in coping with these challenges in an appropriate and integrated way. Unlike ordinary conferences, the MIOS-INTEROP workshop will be a real workshop, providing ample time for discussions, during one and a half day. Consequently, the paper presentations will be short, covering only the highlights and preferably arranged around a number of themes or topics.

We would like to thank all PC members for the valuable work they have done in order to ensure the scientific quality of the selected papers and all the authors

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 406–407, 2005.
© Springer-Verlag Berlin Heidelberg 2005

for their contribution. We hope you will enjoy reading these papers and you will find them valuable for your research and knowledge.

August 2005

Antonia Albani, University of Augsburg
Jan L.G. Dietz, Delft University of Technology
Hervé Panetto, University Henri Poincaré Nancy I
Monica Scannapieco, University of Rome "La Sapienza"
(MIOS+INTEROP'05 Program Committee Co-Chairs)

Registering a Business Collaboration Model in Multiple Business Environments

Birgit Hofreiter and Christian Huemer

Department of Distributed and Multimedia Systems, University of Vienna,
Liebiggasse 4, 1010 Vienna, Austria
{birgit.hofreiter, christian.huemer}@univie.ac.at

Abstract. Today business registries are regarded as means of finding services offered by a business partner. However, business registries may also serve as means of searching inter-organizational business process definitions that are relevant in one's own business environment. Thus, it is important to define in which environments an inter-organizational business process definition is valid. Furthermore, environment-specific adaptations of the business process definition may be registered. In this paper the business process definitions are based on UMM business collaboration models. We discuss two approaches: Firstly, the binding of a model to business environments is specified within the model itself. Secondly, the binding of a model to business environments is defined in the registry meta-data.

1 Motivation

In recent years a lot of interest in inter-organizational systems has been directed toward Web Services in particular and Service-oriented Architectures (SOA) in general. An organization provides services and publishes the interfaces of these services in a registry. In order to facilitate the search for a potential partner the services are classified according to well-known schemes, like NAICS for industries, UN/SPSC for products and ISO 3166 for geographical information. The process of finding and binding services works quite well for atomic services. However, an inter-organizational process is complex requiring composite services on each partner's side to interact with each other in a given choreography in order to realize a common business goal. If business partners develop their interfaces in isolation from each other, it is very unlikely that the interfaces will match each other. This hinders interoperability and organizations are not able to do business electronically.

As a consequence organizations must share a common business process in the collaborative space. UN/CEFACT's Modeling Methodology (UMM) [20] provides a methodology to develop collaborative business process models describing both the choreography in the collaborative space and the data to be exchanged. Furthermore, it exactly defines the necessary interfaces on each partner's side. If an organization wants to do business electronically it has to complete the following steps: (1) Select a business collaboration model. (2) Bind private interfaces to the collaborative process. (3) Register the private interfaces (4) Register the support of a certain role in a business collaboration.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 408 – 420, 2005.
© Springer-Verlag Berlin Heidelberg 2005

In this approach potential business partners are able to find each other by searching for a conjunct role in a supported business collaboration. However, it is required that UMM business collaboration models are available in a registry. In this paper we elaborate on the registration of UMM business collaboration models and their assignment to classification schemes.

2 Related Work

The idea of defining business processes crossing organizational boundaries goes back to ISO's Open-edi reference model [5]. A first implementation of the choreography aspects of this model was a Petri-Net approach contributed by Lee [8]. Also other authors used Petri-Nets to define the workflow between organizations [9,11,24]. Recent approaches consider long-running business transactions in Web Services environments [12,16]. A lot of different standard languages to describe an orchestration or a choreography of a process have been developed. The Business Process Execution Language (BPEL) [1] seems to be the most popular one. Its abstract version supports the definition of inter-organizational business processes [10]. However, BPEL always describes a choreography from the viewpoint of a single partner. This means it is not possible to describe a single choreography for the overall collaboration - as it is required in our approach. XML based choreography languages supporting this requirement are the recently created Web Services Choreography Description Language (WS-CDL) [7] and the ebXML Business Process Specification Schema (BPSS).

Since choreography languages and Web Services are expressed in XML, there have been attempts to model them in a graphical syntax. For this purpose BPMI is developing the Business Process Modeling Notation (BPMN) [25]. This notation presents the amalgamation of best practices in the business process modeling community. Other approaches use UML to visualize Web Services and their choreography [13,17,19]. UML is also used in the standards of RosettaNet [18] and UN/CEFACT's modeling methodology (UMM) [20]. Our paper builds up on the latter one.

Work on registries is dominated by Web Services' UDDI [15] and ebXML's Registry and Repository [14]. In this paper we do not concentrate on the details of how an object, i.e. a business collaboration model, is managed by the registry and its information model. We are more interested in classifying the business objects within a UDDI registry. The classification schemes that seem to be most relevant for our approach is the context driver concept of ebXML core components [22] and UN/CEFACT's catalog of common business processes [23].

3 UN/CEFACT's Modeling Methodology (UMM)

UMM is a methodology for describing inter-organizational business processes. It defines a UML profile - i.e. a set of stereotypes, tagged values and constraints - in order to customize the UML meta model for the special purpose of modeling the collaborative space in B2B. The UMM methodology leads to so-called business

collaboration models. In order to register UMM models the graphical UML syntax must be expressed in a machine-readable format. Currently, UN/CEFACT's business collaboration schema specification (BCSS) project is defining those XML metadata interchange (XMI) flavours that may be used to capture a UMM business collaboration model. Thereby, BCSS makes use of JSR40/ JMI project of the Java Tools Community (JTC) [6].

In the UMM methodology we use three steps that are similar to the first steps in a software development process. In order to demonstrate these steps we use an oversimplified purchase order management example. The resulting artefacts are presented in Fig. 1. The ***business domain view*** is used to gather existing knowledge. Business processes are discovered not constructed. This helps to identify possible collaborations in the next step, the ***business requirements view***. Use cases and associated worksheets are used to collect the requirements of identified business collaborations. Fig. 1a shows a use case diagram for the identified business collaboration *purchase order management* that we use as an example throughout the paper. The business collaboration use case of our purchase order management is built by more basic operations: *register customer*, *request for quote*, and *order product*. This is denoted by the include relationships to the corresponding business transaction use cases.

The ***business transaction view*** covers the analysis model defining the choreography and the information exchanged. It consists of three main artefacts:

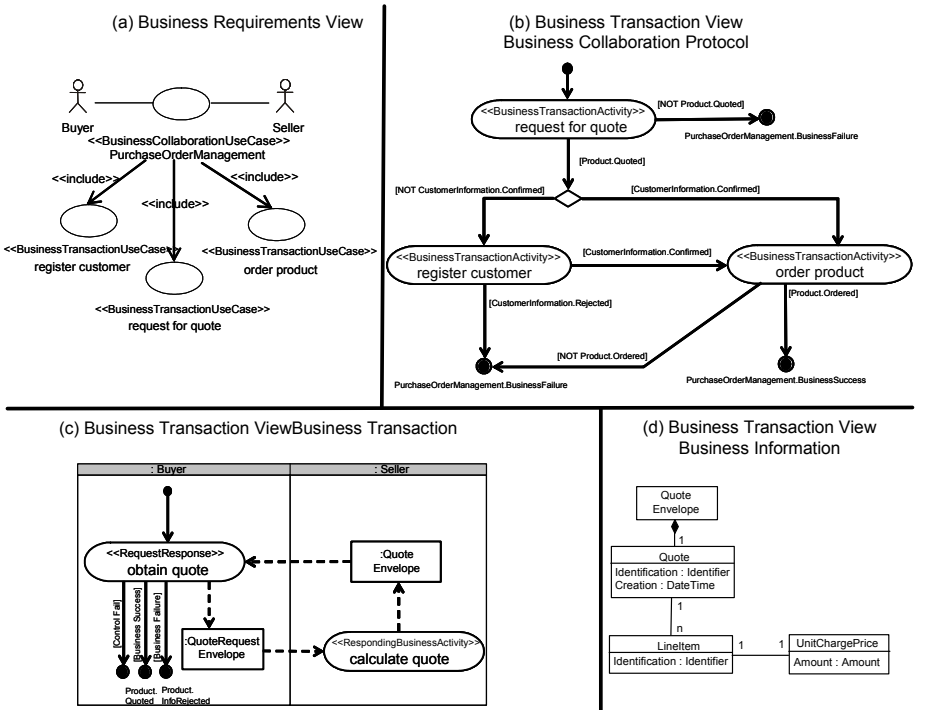


Fig. 1. UMM artefacts for business collaboration: purchase order management

business collaboration protocol, business transaction and business information. Communication in a business collaboration is about aligning the information systems of the business partners. Aligning the information systems means that all relevant business objects (e.g. purchase orders, line items, etc.) are in the same state in each information system. If a business partner recognizes an event that changes the state of a business object, it initiates a business transaction to synchronize with the collaborating business partner. It follows that a business transaction is an atomic unit that leads to a synchronized state in both information systems. A UMM transaction follows always the same pattern due to its strict definition. The *request for quote* business transaction (Fig. 1c) follows this pattern. A business transaction is always performed between two business partners that are assigned to exactly one swimlane each. Each partner performs exactly one activity. An object flow between the requesting and the responding business activity is mandatory. An object flow in the reverse direction is optional. In the *request for quote* business transaction of Fig. 1c the *buyer* performs *obtain quote*, which outputs a *quote request envelope*. This envelope is input to the *calculate quote* activity executed by the *seller*. Since the finale state depends on a decision of the *seller*, a *quote envelope* is returned. Depending on the decision, a product is finally in state *quoted* or *information rejected*. Each of the business transaction use cases of Fig. 1c results in a corresponding business transaction.

The requirements of a business collaboration use case are transformed into the choreography of a so-called business collaboration protocol. Fig. 1b shows the business collaboration protocol for our *purchase order management*. It defines a choreography amongst business transactions, namely *request for quote*, *register customer* and *order product*. It follows that each activity in the business collaboration protocol is refined by the activity graph of a business transaction. For example, the *request for quote* activity in Fig. 1b is refined by the graph of Fig. 1c. The transitions between the activities are guarded by the states of business objects.

Finally, the information exchanged in transactions must be unambiguously defined. Each object in an object flow state is an instance of a class representing an envelope. The aggregates within this envelope are defined in a class diagram. Since UMM is a business state centric approach, the class diagram should only contain information that is needed to accomplish an intended state change. Fig. 1d includes the class diagram for the *quote envelope*, which is exchanged in the business transaction of Fig. 1c. This class diagram is based on ebXML core components [22]. They are assembled and restricted in a way that a business object *product* may be set to state *quoted*, but no additional information is transferred.

4 Internal Binding of Models and Environments

4.1 Binding a Model to Its Environments

It is envisioned that key players in a business domain will develop UMM business collaboration models. These key players may be standard bodies, like UN/CEFACT itself, industry groups, like EAN/UCC, SWIFT, ad-hoc groups for specific processes, or even market leaders in certain domains. Business collaboration models must be

public in order to attract interested parties in the domain. Hence, they are stored in a registry. In this paper, we do not care about the data format used in the registration process - be it a whole model in JMI-compliant XMI [6], or subparts of a model in BPSS [4,21] or WS-CDL [26].

A business collaboration model must be classified in the registry according to its business context. This means that the body who developed the business collaboration must bind it to its business environment. Assume that the model in Fig.1 was developed by the Austrian Tourism Board for travel packages in Austria. For the sake of re-use, a model should be valid in multiple business environments. Assume that the Cyprus Tourism Board finds the Austrian model in the registry and realizes that this model fits the Cyprus needs as well. There is only a little difference: In Cyprus it is only valid for hotels, not for travel packages.

The current UMM proposal provides bindings of a business process model to a business environment within the model itself. In fact there exist two concepts to realize this binding within a UMM model. First of all, the business domain view package in UMM includes sub-packages for business areas. Each business area includes subpackages for process areas. Business processes and business collaboration use cases will be assigned to a corresponding package. Business areas and process areas should refer to the normative categories specified in the UN/CEFACT common business process catalog [23]. This catalog lists eight business areas: procurement/sales, design, manufacture, logistics, recruitment/training, financial services, regulation, and health care. The five process areas correspond to the phases known from Open-edi [5]: planning, identification, negotiation, actualization, and post-actualization. This is a rather rough classification scheme. It is not granular enough to assign a business collaboration to a real world business environment. The left hand side of Fig. 2 shows a resulting package structure within

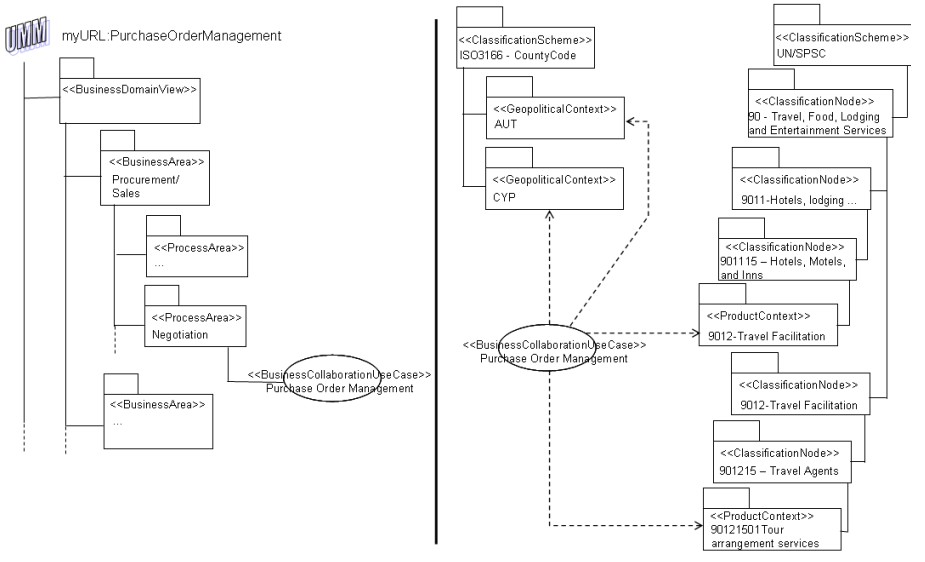


Fig. 2. Binding Business Collaboration Use Cases and Business Environments in UMM

the business domain view of a UMM model and the attempt to assign the business collaboration protocol use case to the correct package. Accordingly, the purchase order management collaboration (of our tourism case) is classified into the business area procurement/sales and, within there, in the process area negotiation.

The second classification concept is based on well accepted classification schemes. However, the realization of this concept is deficient. UMM proposes to create a package structure for each classification scheme and its sub-nodes. Dependency relationships from the business collaboration use case to all appropriate classification packages are created. The right hand side of Fig. 2 depicts the classification of our purchase order management according to the UN/SPSC product classification and the ISO 3166 country code. The UN/SPSC uses four levels of classification nodes: segment, family, class, and commodity. In the ISO 3166 only one level of classification exists. The purchase order management depends on the geopolitical contexts of *Austria* and *Cyprus*, and on the product contexts of *tour arrangement services* and *hotels*. By this example two major drawbacks are becoming obvious: Firstly, the package-based approach does not allow category groups. This means the business collaboration is now valid in both countries in both product contexts - which is not correct. Instead one would need a category group for *Austria* and *tour arrangement services* and another group for *Cyprus* and *hotels*. The second drawback is the overwhelming package structure representing the code lists. All the packages are basically empty. They are just created for the sake of establishing dependency relationships.

As a consequence we prefer another mechanism to bind a business collaboration to business environments. We use tagged values instead of packages. This means a tagged value for the business environment is added to the stereotype of a business collaboration use case in the UMM meta model. The business environment is defined by a business context statement that is based on the eight context categories defined by ebXML core components to describe a business environment [22]. The business context statement is a boolean expression connecting business context descriptors:

BusinessContextStatement :: =
[<BusinessContext> [<BooleanOperator> <BusinessContextStatement>]? |
[(<BusinessContext> <BooleanOperator> <BusinessContextStatement>)]

BusinessContext ::= <BusinessContextDriver> <relationalOperator> “<literal>”

BusinessContextDriver ::= **BusinessCollaboration** | **BusinessTransaction** |
ProductClassification | **IndustryClassification** | **Geopolitical** |
Official Constraints | **BusinessProcessRole** | **SupportingRole** | **SystemCapabilities**

BooleanOperator ::= **AND** | **OR** | **XOR** relationalOperator ::= **=** | **>** | **<** | **>=** | **<=** | **<>**

Accordingly, the business environment tagged value of the purchase order management use case will have the following value:

(Geopolitical = "AUT" AND ProductClassification = "Tour arrangement services") OR
(Geopolitical = "CYP" AND ProductClassification = "Hotels")

4.2 Business Environment-Specific Variations

By now, we are able to define that a business collaboration is applicable in multiple business environments - however only if the collaboration is identically executed in each business environment. A slight variation would require the creation of a new

collaboration. Imagine the slight difference in our tourism example for the request for quote transaction (Fig. 1c): In the Austrian case the response time for receiving the quote is 24 hours and authorization as well as non-repudiation are required. In the Cyprus case the response time is just 1 minute and the security requirements do not apply.

The response time and the security parameters mentioned above are specified in the tagged values of a requesting business activity in a business transaction. Each tagged value carries exactly one value in the original UMM approach. In this case it is impossible to include business context variations. To overcome this situation the tagged value may include an if-statement for the context variations. In this case, the *if*-clause includes a business context statement as described above and the *then*-clause sets the value for the corresponding business environment. Further, the *if*-statement may include *elsif*-clauses and an *else*-clause.

The tagged value for *time to perform* in our tourism example is specified as stated below. In this statement we assume a default value of *12 hours*, which is neither used in the Austrian nor in the Cyprus case. The boolean values for *is authorization required* and *is non repudiation required* are set by similar if-statements.

```
if (Geopolitical = "AUT" AND ProductClassification = "Tour arrangement services")
      then "PT24H"
elsif (Geopolitical = "CYP" AND ProductClassification = "Hotels")
       then "PT1M"
else "PT12H"
```

Of course, such variations do not only apply to the requesting business activity (cf. [3]). In business transactions, variations exist for the responding business activity and the security parameters of the object flows. In the business collaboration protocol, the tagged values of business transaction activities may differ from environment to environment. Transitions may only be valid in certain environments, or the business states guarding the transition may differ. Finally, the business information exchanged in a transaction may be different. This last variation is tackled by the core components technical specification [22].

5 External Binding of Models and Environments

5.1 Binding a Model to Its Environments

In the previous section we defined the binding of a business collaboration model to business environments within the model itself. Thus we call it an internal binding. In this section we concentrate on an external binding. The definition of the applicable business environments are not defined within the model itself, but in the registry. Similarly the differences between business environments must be handled by registry meta-data and not within the model.

The motivation for an external binding is best demonstrated by the simple example we used already before. Imagine the Austrian tourism board developed its purchase order management for travel packages. It registers its model. The Cyprus tourism board detects this model in the registry and finds it appropriate for their hotel domain. In the case of an internal binding, the Cyprus tourism board must change the model in

order to declare it as valid in their environment and to define the context variations. This means a new version of a business collaboration model is created whenever a new domain is interested in it - even if the basics of the model do not change. If the Austrian board wants to update the model later, this may have consequences on versions created by other organizations. This means a complex version mechanism is necessary. Maintenance would be much simpler in the case of external bindings: A business collaboration model is defined independently of the business environment. Metaphorically, each domain organization that accepts the model for their environment puts a sticker on the model. The Austrian tourism board creates the model and puts a sticker on it. Later on, the Cyprus board finds the model and puts another sticker with the Cyprus-specifics on it. If the Austrian Tourism board updates the model it creates a new version, and moves the sticker from the old to the new version. It is up to the Cyprus board to leave the sticker on the old version or to move it to the new one as well.

Of course we do not have any stickers in the electronic world. Instead of stickers, this concept must be realized by the registry meta-data assigned to a business collaboration model. All four main UDDI data structure types provide a structure to support attaching categories [15]. One of these data types is the *tModel*. *tModels* are used to engender the notion of shared specifications. Inasmuch a *tModel* contains the address where a business collaboration model can be found. The *tModel* structure allows to attach a *category bag*. A *category bag* may include *keyed reference groups*. Each group is used to describe a business environment. The following *tModel* is used to bind our purchase order management to Austrian travel packages and Cyprus hotels:

```
<tModel tModelKey="uddi:whoever:umm:purchaseordermanagement">
    <name>http://www.whoever.org/purchaseOrderManagement</name>
    <categoryBag>
        <keyedReferenceGroup>
            <keyedReference
                tModelKey="uddi:uddi.org:ubr:categorization:unspsc"
                keyName="UNSPSC:Tour arrangement services"
                keyValue="90.12.15.01"/>
            keyedReference
                tModelKey="uddi:uddi.org:ubr:categorization:iso3166"
                keyName="GEO:Austria"
                keyValue="AT"/>
        </keyedReferenceGroup>
        <keyedReferenceGroup>
            <keyedReference
                tModelKey="uddi:uddi.org:ubr:categorization:unspsc"
                keyName="UNSPSC:Tour arrangement services"
                keyValue="90.12.15.01"/>
            keyedReference
                tModelKey="uddi:uddi.org:ubr:categorization:iso3166"
                keyName="GEO:Austria"
                keyValue="AT"/>
        </keyedReferenceGroup>
    </categoryBag>
</tModel>
```

5.2 Business Environment-Specific Variations

So far, we are able to bind a business collaboration model and business environments in the registry meta-data, if the business collaboration model is identical in each environment. Next we have to consider variations in the execution of business process models. We again use the example of a different response time and security parameters in the *request for quote* business transaction to illustrate the approach.

We are facing now the problem that the concepts that may vary - tagged values, transitions, guards - are part of the UMM model itself. Nevertheless we want to specify variations to these concepts without changing the model itself. Hence, a UMM model includes the defaults for these concepts. In our example, the Austrian tourism board registers the model with the default values. For example, the tagged value of *time to perform* of *obtain quote* is *24 hours*. Later the Cyprus tourism board wants to define that the respective *time to perform* is only *1 minute* in their scenario. However, it cannot change the tagged value in the model itself. For this purpose we develop an XML schema to define business collaboration variations. The code below shows an extract of the schema that we need in our example. The root element *business collaboration variation* uses an *id* attribute to reference the original business collaboration model. The elements beneath are used to specify the variations within this model.

```
<element name="BusinessCollaborationVariation">
    <complexType>
      <sequence>
          <element ref="BusinessCollaborationCharacteristics" minOccurs="1" axOccurs="unbounded"/>
      </sequence>
      <attribute name="baseBusinessCollaborationModelId" type="anyURI"/>
       <attribute name="BaseBusinessCollaborationModelName" type="string"/>
</element>
<element name="BusinessCollaborationCharacteristics">
    <complexType>
      <sequence>
        <element ref="BusinessTransactionActivityCharacteristics"minOccurs="0"maxOccurs="unbounded"/
        <element ref="TransitionCharacteristics" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="BusinessTransactionCharacteristics" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
        <attribute name="nameId" type="anyURI"/>
        <attribute name="name" type="string"/>
        </complexType>
</element>
<element name="BusinessTransactionCharacteristics">
   <complexType>
     <sequence>
       <element ref="RequestingBusinessActivityCharacteristics" minOccurs="0"/>
       <element ref="RespondingBusinessActivityCharacteristics" minOccurs="0"/>
       <element ref="RequestingBusinessInformationCharacteristics" minOccurs="0"/>
       <element ref="RespondingBusinessInformationCharacteristics" minOccurs="0"/>
     </sequence>
      <attribute name="nameId" type="anyURI"/>
      <attribute name="name" type="string"/>
    </complexType>
</element>
```

```
<element name="RequestingBusinessActivityCharacteristics">
    <complexType>
        <attribute name="timeToPerform" type="duration"/>
        <attribute name="timeToAcknowledgeReceipt" type="duration"/>
        <attribute name="timeToAcknowledgeAcceptance" type="duration"/>
        <attribute name="isAuthorizationRequired" type="boolean"/>
        <attribute name="isNonRepudiationRequired" type="boolean"/>
        <attribute name="isNonRepudiationReceiptRequired" type="boolean"/>
        <attribute name="isIntelligibleCheckRequired" type="boolean"/>
        <attribute name="retryCount" type="integer"/>
          </complexType>
</element>
```

The Cyprus tourism board creates the following instance of this XML schema. The *id* of the business collaboration variation points to the original business collaboration model. It redefines the characteristics of the business collaboration *purchase order management*. Within this business collaboration it redefines the characteristics of the business transaction *request for quote*. In this transaction it changes the tagged values for *time to perform*, *is authorization required* and *is non repudiation required* of the requesting business activity. It is clear that this is the *obtain quote* activity, because a business transaction includes always exactly one requesting business activity.

```
<BusinessCollaborationVariation
    baseBusinessCollaborationModelId="http://www.whoever.org/purchaseOrderManagement">
     <BusinessCollaborationCharacteristics name="purchaseOrderManagement">
          <BusinessTransactionCharacteristics name="requestForQuote">
               <RequestingBusinessActivityCharacteristics
                    timeToPerform="PT1M" isAuthorizationRequired="false"
                    isNonRepudiationRequired="false"/>
          </BusinessTransactionCharacteristics>
     </BusinessCollaborationCharacteristics>
</BusinessCollaborationVariation>
```

Once the Cyprus board has created this business collaboration activity, it is able to create a *tModel* that refers to this variation. This means the Cyprus board does not add another keyed reference group to the *tModel* of the original model. Instead it creates another *tModel* for the XML Schema of the variation which links to the original business collaboration. The *tModel* of the variation includes a category bag that includes the classification of the Cyprus case. If afterwards another Tourism board comes along, that wants to use the classification exactly as the Cyprus board specified, it adds its classification to the category bag - which must then include *keyed reference groups* again.

```
<tModel tModelKey="uddi:someoneelse:umm:purchaseordermanagementvariation">
    <name>http://www.someoneelse.org/purchaseOrderManagementVariation</name>
     <categoryBag>
          <keyedReference
               tModelKey="uddi:uddi.org:ubr:categorization:unspsc"
               keyName="UNSPSC:Tour arrangement services"
               keyValue="90.12.15.01"/>
```

```
            keyedReference
                tModelKey="uddi:uddi.org:ubr:categorization:iso3166"
                keyName="GEO:Austria"
                keyValue="AT"/>
            </keyedReferenceGroup>
        </categoryBag>
    </tModel>
```

6 Summary

Inter-organizational business processes are usually quite complex. The acquisition of a product from a business partner is not a one step process. Usually a lot of communications between the business partners are necessary. Electronic communication between the applications of the business partners requires an unambiguous choreography and unambiguous information exchanged. If each organization develops its interfaces independently from each other, it is rather unlikely that they will inter-operate.

Hence there is a need for shared business collaboration models, which exactly define each roles behavior and to which each partner can bind its private processes. UMM provides a methodology to develop these business collaboration models. These business collaboration models must be publicly available. Therefore, this paper deals with the registration of UMM models. We focus on the registration of the models independent of the support by business partners. This means, we do not concentrate on the binding of organizations to roles in business collaboration models.

Organizations will search a registry to find an appropriate model. A search will look for models that are valid in a business environment of interest. Models must be bound to specific business environments. Thus, we elaborated on the binding of a model to one or more business environments. Although different business environments share a common model, the execution may be slightly different. So we focused also on the registration of these slight variations.

Both the binding and the variations may be specified either in the UMM model itself or in the registry meta-data. In the current UMM approach a binding is defined within the model itself. An evaluation of the current approach showed that it is insufficient because it does not support category groups and it results in an overloaded package structure. We developed an alternative approach based on tagged values. The current UMM does not allow the definition of slight variations. Thus we extended the UMM to allow variations for tagged values, transitions and transition guards.

In contrary to the current UMM vision of defining a binding within the model itself, we investigated in a binding of the model and its environment externally in the registry meta-data. We prefer this approach because the maintenance of the bindings does not effect the models. This results in a much simpler versioning mechanism. We demonstrated by the means of UDDI that this binding mechanism is available of today. ebXML registries would be prepared as well. More complicated is the external definition of variations. For this purpose we had to develop an XML schema that references the original model and captures the variations. Furthermore, we demonstrated its usage within a registry by the means of UDDI again.

References

1. Andrews, T., et al.: Business Process Execution Language for Web Services, V. 1.1. (2003) http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbizspec/html/bpel1-1.asp
2. Eisenberg, B., Nickull, D.: ebXML Technical Architecture Specification, v1.0.4. (2001) http://www.ebxml.org/specs/ebTA.pdf
3. Hofreiter, B., Huemer, C., Winiwarter, W.: OCL-Constraints for UMM Business Collaborations. Proc. of 5th Int'l Conf. on Electronic Commerce and Web Technologies (EC-Web 2004), Springer LNCS, (2004)
4. Hofreiter B., Huemer, C., Kim J.-H: Choreography of ebXML business collaborations. accepted and to appear in: Journal of Information Systems and e-Business. Springer (2005)
5. ISO: Open-edi Reference Model. ISO/IEC JTC 1/SC30 ISO Standard 14662 (1997)
6. Java Community Process: JSR-000040 The JavaTM Metadata Interface (JMI) Specification. http://jcp.org/aboutJava/communityprocess/final/jsr040/index.html
7. Kavantzas, N. et al: Web Services Choreography Description Language, Version 1.0. W3C (2004) http://www.w3.org/TR/ws-cdl-10
8. R.M. Lee: Documentary Petri Nets: A Modeling Representation for Electronic Trade Procedures. In: Business Process Management, Models, Techniques, and Empirical Studies. Springer LNCS, Vol. 1806 (2000) 259 - 375
9. Lenz, K., Oberweis, A.: Interorganizational Business Process Management with XML Nets. In: Advances in Petri Nets. Springer LNCS, Vol. 2472 (2003)
10. Leymann, F., Roller, D., Schmidt, M.-T.: Web Services and Business Process Management. IBM Systems Journal, Vol. 41, No. 2, 2002
11. Ling, S., Loke S.W.: Advanced Petri Nets for Modelling Mobile Agent Enabled Interorganizational Workflows. Proc. of 9th IEEE Int'l Conf. and Workshop on the Engineering of Computer-Based Systems (ECBS 2002), IEEE Computer Society (2002)
12. Little, M.: Transactions and Web Services. Communications of the ACM, Vol. 46, No. 10 (2003) 49 - 54
13. Mantell, K.: From UML to BPEL - Model Driven Architecture in a Web Services World. IBM Developer Works (2003), http://www-128.ibm.com/developerworks/webservices/library/ws-uml2bpel/
14. OASIS: ebXML Registry Information Model v2.0. (2001) http://www.ebxml.org/specs/ebrim2.pdf
15. OASIS: UDDI Version 3.0.2. http://uddi.org/pubs/uddi_v3.htm
16. Papapzoglou, M.P.: Web Services and Business Transactions. WWW, Internet and Web Information Systems, Vol. 6, Kluwer (2003) 49-91
17. Provost, W.: UML for Web Services.XML.com (2003) http://www.xml.com/lpt/a/ws/2003/08/05/uml.html
18. RosettaNet: RosettaNet Implementation Framework, Core Specification V02.00.01. (2002) http://www.rosettanet.org/rnif
19. Thšne, S., Depke, R., Engels, G:: Process-Oriented, Flexible Composition of Web Services with UML. Int'l Workshop on Conceptual Modeling Approaches for e-Business: A Web Service Perspective (eCOMO 2002), (2002)

20. UN/CEFACT: UMM Meta Model, Revision 12; (2003) http://www.untmg.org/downloads/General/approved/UMM-MM-V20030117.zip
21. UN/CEFACT: ebXML - Business Process Specification Schema v1.10. (2003) http://www.untmg.org/downloads/General/approved/ebBPSS-v1pt10.zip
22. UN/CEFACT: Core Components Technical Specification, Version 2.01. (2003) http://www.unece.org/cefact/ebxml/CCTS_V2-01_Final.pdf
23. UN/CEFACT: Common Business Process Catalog, v2.0 (2005) Not published yet (link will be included in final version)
24. van der Aalst, W.M.P.: Interorganizational Workflows: An Approach based on Message Sequence Charts and Petri Nets. Systems Analysis - Modelling - Simulation, Vol. 34, No. 3 (1999) 335-367
25. White, S: Business Process Modeling Notation Working Draft , V 1.0. (2003) http://www.bpmi.org/bpmn-spec.esp
26. W3C: Web Service Choreography Interface V 1.0. (2002) http://www.w3.org/TR/wsci/

Component Oriented Design and Modeling of Cross-Enterprise Service Processes

Rainer Schmidt

Department of Computer Science, University of Applied Sciences,
Beethovenstraße 1, 73430 Aalen
Rainer.Schmidt@fh-aalen.de

Abstract. Service processes are an important kind of cross-enterprise business processes. However, they show particular properties which require new approaches of design and modeling. Therefore, in this paper a method for designing and modeling service processes is developed. It is component-oriented and uses so-called perspective elements as granularity of the components. The service processes created by the method are both aligned to the customer requirements and efficient in their operation by the use of standardized components.

1 Introduction

The providing of services plays a more and more important role in modern and internationally networked economies. Service processes cross enterprises and use the resources of multiple organizations. Therefore service processes are an important kind of cross-enterprise processes. They are established to provide services such as support for computer systems, customer care etc. However, for example in the area of information technology services (IT-services) only little attention was paid to their design. Therefore it does not surprise, that the providing of services is far away from the efficiency known from production systems in mature industries or software development. Now, standards for IT service processes such as the Information Technology Infrastructure Library (ITIL) [ITIL], is spreading quickly and has become the de-facto-standard for service management and service processes [Schm04]. However, in spite of this progress there are also some shortcomings. Most important, ITIL provides not method which is both efficient and allows creating processes individually tailored to customer requirements.

Therefore, this paper will present a component oriented design and modeling method for service processes offering both efficiency and individual solutions according to customer requirements. It enables the "industrialized" production of services. The paper will proceed as follows. In section 2, the properties of service processes and especially the differences to ordinary business processes will be analyzed. The component-oriented method for designing and modeling Service-processes is defined in section 3. Important parts of the method are presented in the following four sections in more detail. In section 4 a framework of requirements for Service processes is defined. The granularity of the components is discussed in section 5. The service component repository is introduced in 6. The creation of composite services is described in section 7. Finally, related work is discussed in section 8.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 421 – 430, 2005.
© Springer-Verlag Berlin Heidelberg 2005

2 Service Processes

To clarify the characteristics of service a process, a case study is used which origins from the ITIL module incident management / service desk [ITIL]. The service process is a three level IT-support for problems of a computer system .The IT-support process is operated by a service provider in the building of the customer. The three level user support consists out of a service desk at level 1, a team of specialists at level 2 and third-party specialists at level 3. All support levels interact with the customer to analyze the problem and they access the customer's computer system to configure it. Furthermore, each level has a determined reaction time for requests of the customer. The service desk at level 1 is the primary point of contact for the customer's staff. The service desk has to react to incidents within 10 minutes. Many problems can be solved by the service desk. Only if a problem cannot be solved by the service desk, it is forwarded to the second level support. The second level support has to react within two hours. But there are also problems, which can not be solved by the second level support. These problems are forwarded to specialists of external service providers who are the third level support. They have to react within one day.

Service processes have many properties in common with ordinary business processes. However, service processes are also products and also have to fulfill requirements, which are typical of products: Therefore, service processes have to fulfill the individual requirements of the customer and are to be offered at a competitive price. Another characteristic of service processes is their high degree of division of labor. There are many interactions between the service provider and the customer and third party service providers. Both have to be integrated during the whole process and not only at the beginning and the end of the process: In the example above, the customer has to be interrogated for further details of his incident report. Advice is sought from the third level support.

Service processes differ from traditional business processes also because they extensively use external resources both from the customer and third party service providers. External resources have to be appropriately obtained, integrated and administered. For example, before configuring the customer's computer system, one has to have administrative privileges to do so. In addition, if external resources are no longer available but needed for service providing, a procedure to correct these errors has to be started. Finally resources of the customer which have been used for service providing have to be given back at the end of the service providing.

Not only the execution but also the potential to execute the service process is important to the customer. In the example above, it is important for the customer that his staff may call the service and start the service within a predefined reaction time. Therefore service providers have to make available a predefined potential to perform a service process. This potential is measured as service level. To reach a certain service level, resources have to be kept available, as services cannot be kept in store as material products. In the example, one has to have ready properly trained staff available in the service desk, regardless whether there are calls or not.

3 Design and Modeling of Service Processes

Service processes both have to be flexibly adaptable to the customer's requirements and to be offered at a competitive price. At first sight this creates a dilemma: if

services are individually developed to the customer's specification, they become more expensive because there are no scaling effects. Otherwise, if standardized services are used, for example based on ITIL, there are scaling effects but it is not possible to meet the individual customer's requirements. A common way to escape from this dilemma is the use of components. A product is composed of standardized components in a way individually specified by the customer. By choosing from a multitude of components, both the efficiency from mass production and the individual solution to the customer can be achieved. Thus, the use of components is the key to the "industrialization" of service providing. By composing services out of prefabricated components, the providing of services gains much in efficiency and flexibility as "classical" industries already have achieved.

3.1 Component Orientation

As service processes are immaterial artifacts, it can help to look for already existing component concepts for immaterial artifacts. One of the most important areas where immaterial artifacts are used is software engineering. There the term component was first used in [McIl69]. In the last years, the concept of software component got more and more important in developing flexible software at a minimum cost [Szyp97]. Systems built of components can be adapted to changed requirements more easily. Core concept of a component is strong encapsulation [Szyp98] so that all context-dependencies [CiSc96] of the component are represented in an explicitly defined interface and the implementation of the component is not visible outside.

3.2 Component Oriented Design and Modeling of Service Processes

The component oriented design of service processes starts with the elicitation and analysis of the (informal) customer requirements. The formalized requirements are used to retrieve service components from a component repository. This repository is not static but can easily be extended by additional components, which may be independently developed, for example by a subcontractor. In a gap analysis, the potential solution using the retrieved components is compared to the original requirements definition. Gaps can be handled in three ways: First, the original requirements definition can be renegotiated with the customer. This may lead to tradeoffs as discussed in [AlFC01]. Second, the creation of new components can be initiated to fill the gaps. Third, the gaps can be closed by specialization, as described later. The next step is the selection of components. The automation of the selection process is highly dependent on the availability of a domain-specific ontology to specify the component functionality. Finally the selected components are specialized and composed to a composite service process. The specialization is done by adapting the component to individual requirements without changing their external interfaces and behavior. The encapsulation of the component by its interfaces impedes the visibility of internal changes. Furthermore, the strong interfaces make explicit the context-dependencies of the component.

In the following sections, the component oriented design and modeling of service processes is described in more detail. First, a framework for requirements elicitation and analysis is defined in section 4. The granularity of the components in the

component repository is discussed in section 5. The structure of the components and the component repository is defined in section 6. Finally, the specialization and composition of service components to composite services is described in section 7.

4 Framework of Requirements for Service Processes

To assure the completeness of requirements elicitation and analysis, a framework is necessary, which classifies all requirements on an abstract level. The requirements framework also creates a domain specific ontology necessary for an appropriate retrieval and selection of components: It also allows retrieving and selecting the most appropriate component during component selection. Frameworks for requirements classification already exist for the design of workflows [JaBu96] and business processes [BKKR03]. Frameworks for business components are defined in [Turo01] and [Over03]. The framework in [BKKR03] comprises eight perspectives, specifically functional, operational, behavioral, informational, organizational, causal, historical, and transactional. The functional perspective describes what the process has to do, which input it uses and which output it creates and the decomposition of the process into activities. The operational perspective complements the functional perspective by describing how the tasks specified in the functional perspective are done. Thus it describes the implementation. The behavioral perspective defines, when and under which preconditions activities are performed. In the informational perspective the information to be exchanged between activities is defined. The organizational perspective describes who participates in the process in which roles. The causal perspective describes why something has to be done in a process. The historical perspective is used to represent the change of process instances in time. In the transactional perspectives the clustering of activities into transactions is defined.

However, with the analysis of service processes of section 2 in mind, it is clear that the framework of requirements for service processes needs some additional perspectives. First, there is the need for a perspective to describe the interactions between the participants in the service processes. These interactions may be a simple one-way communication, bidirectional or follow complex protocols. Interactions can be further described by the following properties: First, the start of the interaction may be automatic or on user initiation. Second, the participants can be predetermined or have to be decided in an ad hoc manner. Also the participation of mediators, which are not member of the participating organizations, may be necessary, for example to settle a dispute. Third, the interaction may have a definitive structure and the end of the interaction is determined during the interaction. Finally, the interaction can be differentiated if they have a defined outcome or not.

The second additional perspective is the resource perspective. Service processes extensively use external resources both from the customer and third party service providers which have to be appropriately integrated and administered. Therefore the resource perspective specifies not only the resource itself, but also the procedure needed to obtain and return the resource, for example from the customer. Furthermore an error handling procedure is necessary, if the availability and reliability of the resource is not as defined.

The third perspective, the service level perspective, is needed to define the potential to perform activities. It describes the rights and duties for the customer and the service provider, the service performance indicators (SPIs), the measurement of the service performance indicators and change procedures

5 Component Granularity

An important problem in creating a component-oriented design method is the granularity of the components. On one hand, it has to be assured, that changes to service processes affect a minimal number of components and the configuration effort is minimized. On the other hand, there must be enough possibilities for configuration but the effort may not become too big. If there are not enough configuration possibilities the value to the customer is reduced. Also, one has to avoid, that the component granularity predetermines a solution not optimal for the customer. Another risk is the creation of redundancies between components.

The perspectives of the framework are the key for defining an appropriate component granularity. First, components should contain functionality of only one perspective because perspectives evolve independently. For example, the organizational structure of service processes can be changed completely whereas the operational perspective remains unchanged. However, choosing the whole perspective as granularity would not deliver the optimal solution as processes do not contain all elements of a perspective. Therefore, elements of the perspectives should be used as granularity, for example, the interaction type a, a single control flow construct such as fork or a single data type. By choosing elements of perspectives, a maximum of context-independence and encapsulation is achieved. Applying these considerations to

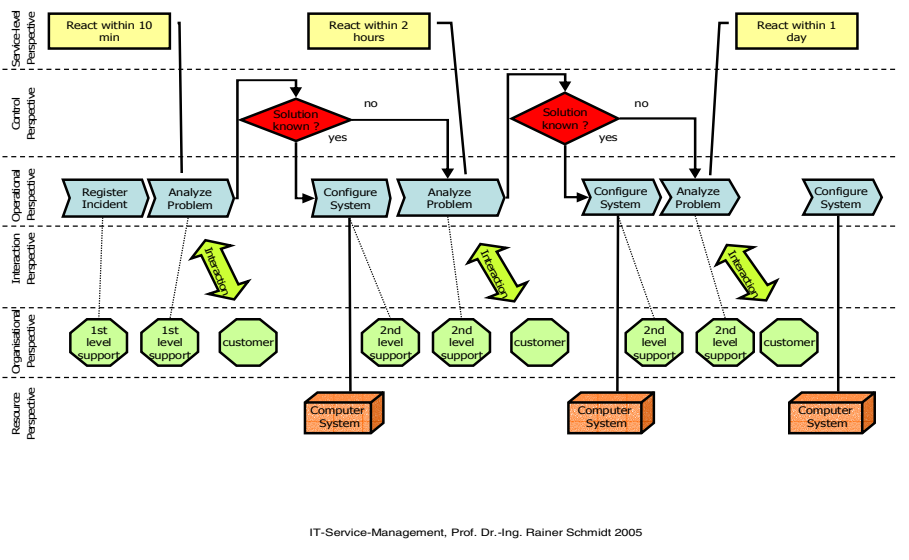


Fig. 1. Perspective-separated case study

the case study, we get the representation as shown in Fig. 1. Here the service process is split up into perspectives and perspective elements. Each perspective is shown as separate layer. (Not all perspectives are shown for the clarity of the drawing. The informational, causal, transactional and historical perspectives are not shown).

6 The Service Component Repository

Based on the considerations above, the repository is partitioned into the perspectives identified in the framework described above. Each perspective is divided into elements. One or several components are associated to each perspective element. All components of a perspective element have the same interfaces in common as specified by the perspective element. The interfaces define the communication with other perspective elements. There are both ingoing and outgoing interfaces. Multiple components are differentiated by their metadata.

7 Creation of Composite Service Processes

A composite service process is created by a set of specialized service components. The specialization of components is done by applying specialization information to the components: the component is parameterized and connected to other components depending on the requirements elicited before. Therefore, the specialization information contains both parameterization and connection information. The parameterization information adapts the component to the individual needs of the composite service. The connection information contains the connections of the component with other components in the context of the composite service. For different composite services there is different specialization information. The specialization information for different composite services is discriminated by the so-called global context identifier. The so-called local context identifier differentiates multiple uses of the same service in a composite service.

An example is given in Fig. 2. There are two composite services A and B. They are created using three components c1, c2 and c3. Composite service A is created by using components c1 and c2 with the specialization information denoted with the context identifier i_{a1}. The identifier is composed from a global and a local context identifier. The global context identifier i_a associates the specialization information with the composite service A. The local context identifier denoted "$_1$ " determines the context within composite service A. The specialization of c1 contains the connection information representing the connection between c1 and c2. Furthermore c1 and c2 are parameterized. Composite service B is created by using components c2 and c3 using specialization information denoted with the global context identifier i_b. By evaluation of the global context identifier, component c2 can differentiate if it is called in the context of composite service A or composite service B. Component c3 is multiply used within composite service B. Therefore there are two sets of specialization information denoted with the global identifier i_b and differentiated by the local identifiers "$_1$ " and "$_2$ " . If c3 is used for the first time, the specialization information denoted with i_{b1} is used. The second time, the specialization information i_{b2} is used.

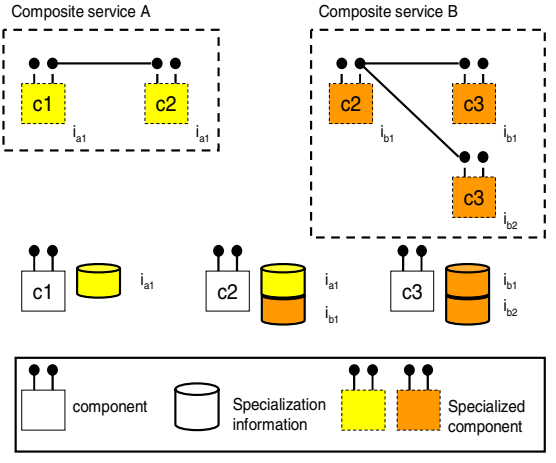


Fig. 2. Composite services

8 Related Work

There are multiple areas of related research. These are approaches to increase the reuse of reference models, design methods for reference models, reuse mechanisms for workflows and business processes, the reuse of e-services and business component specification.

First, there are approaches to increase the reuse in reference models. An in-depth analysis is done in [FeLo02], which shows, that only a component-oriented approach provides both reuse and flexibility: Six classes of methods for the reuse of reference models are differentiated according to their reuse mechanisms. The composition of a service process out of predefined components is thus not possible. Pattern-language oriented approaches such as [Wolf01] define a multitude of dependencies between different models. However, the definition of an appropriate component granularity and suitable interfaces is omitted. Catalog-based methods ([FeLo02], and knowledge based methods ([Kram99], [Schu01]) are aimed at the retrieval of already defined reference models as a whole. No method how to appropriately design components for reuse is presented. The library based approaches [Lang97] store (parts of) reference models in a library using a classification or type system. However, no recommendations for an appropriate granularity of the library entries are given. Instead the formalization of the library elements is focused as in [LaBo97].

Another area of related research is the construction of reference models. In [Broc04] the construction of reference models in a distributed environment is shown. However, only an abstract method to design components for a reference model is given, which is not directly applicable to service processes. In [BöJK03] a modularization approach for services in the information technology business is proposed. It provides a modularization approach but does not use components as context-independent units.

The third area of related work is the reuse of workflow and business process definitions. The reuse and extension of standard business process components is

proposed in [HiLe98]. Reuse of workflow definitions is defined in [WLMP04] using a workflow definition language in [BlWM03]. A component-oriented approach for standard business processes is presented in [Schm03]. Although these approaches contain some interesting points, they do not consider the special properties of service processes identified in section 7.

The specification of business components and the construction of business components frameworks is discussed in [FeTu99], [Over03]. These approaches have a strong focus on standard business processes and are not so easily extensible as [BKKR03]. Questions of component granularity and component identification are covered in [AlDZ05], [ABTW04] for example. There a more top-down approach for component identification is used compared to the bottom up approach presented here.

9 Summary and Outlook

Service processes are an important group of cross-enterprise business processes. They have special properties which require new methods of design and modeling. Service processes are at the same time processes and products and therefore have to be flexibly adaptable to the customer's requirements while being offered at a competitive price. Furthermore service processes show a high degree of interaction with external participants such as the customer and subcontractors. Another difference to standard business processes is the integration of external resources, for example the customer's computer system into the process. Finally, service processes not only have to produce a defined process output but they have also to provide a defined potential to provide the process output called service level.

To cope with these requirements a component oriented design and modeling method using a repository for service processes has been developed. Based on it composite service processes are built from components in a way individually specified by the customer. A high degree of individualization can be achieved, by choosing from a multitude of components. By using a component oriented approach both efficiency and the individual solution to the customer can be achieved. The method presented enables the "industrialized" production of services and can be used to fix the deficiencies of ITIL and thus fully profit from the best practices contained in ITIL. Therefore, further work will apply the concepts developed to ITIL in the German chapter of the ITSMF [ITSMF]. Key points will be the design of appropriate metadata to describe the components and to make them retrievable in the repository. Further research will cover the following topics. First, in a distributed environment, components may use different model representations such as event oriented process chains versus Petri nets. Therefore an integration mechanism has to be developed. The distributed environment may also require the distribution of the component repository. Second, a mechanism for revising and creating versions of the components has to be developed. Third, market mechanisms have to be created to exchange components.

References

[AlFC01] Alves, C., Filho, J. B. P., Castro, J., Analysing the Tradeoffs Among Requirements, Architectures and COTS Components *IV Workshop on Requirements Engineering*, Buenos Aires, Argentina, November 2001.

[AlDZ05] Albani, A.; Dietz, J. L. G.; Zaha, J. M.: Identifying Business Components on the basis of an Enterprise Ontology. Interop-Esa 2005 - First International Conference on Interoperability of Enterprise Software and Applications. Geneva, Switzerland 2005

[ABTW04] Albani, A.; Bazijanec, B.; Turowski, K.; Winnewisser, C.: Component Framework for Strategic Supply Network Development. In: Benczúr, A.; Demetrovics, J.; Gottlob, G. (Hrsg.): 8th East European Conference on Advances in Databases and Information Systems (ADBIS-04), 22 - 25 September 2004, LNCS 3255. Budapest, Hungary 2004, S.67-82.

[BKKR03] M. Bernauer, G. Kappel, G. Kramler, W. Retschitzegger, Specification of Interorganizational Workflows - A Comparison of Approaches, Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2003),

[BlWM03] Blin, J. J.; Wainer, J.; Medeiros, C. B.: A reuse-oriented workflow definition language. International Journal of Cooperative Information Systems. Vol. 12, no. 1, pp. 1-36. Mar. 2003

[BöJK03] Böhmann, T.; Junginger, M.; Krcmar, H. 2003. Modular Service Architectures: A Concept and Method for Engineering IT Services. Paper presented at the 36th Annual Hawaii International Conference on System Sciences (HICSS-36), January 6-9, 2003, Big Island, Hawaii

[Broc03] Brocke, Jan vom: Referenzmodellierung. Gestaltung und Verteilung von Konstruktionsprozessen. Berlin 2003.

[Broc04] Brocke, Jan vom; Buddendick, C.: Organisationsformen in der Referenzmodellierung - Forschungsbedarf und Gestaltungsempfehlungen auf Basis der Transaktionskostentheorie. In: Wirtschaftsinformatik 46 (2004) 5, S. 341-352.

[CiSc96] O. Ciupke, R. Schmidt: Components As Context-Independent Units of Software. WCOP 96, Linz 1996. In Special Issues in Object-Oriented Programming, Workshop Reader of the 10th European Conference on Object-Oriented Programming ECOOP96, Dpunkt Verlag, Heidelberg, 1996, S. 139 – 143.

[FeTu99] Fellner, K.; Turowski, K. (1999): Component Framework Supporting Inter-company Cooperation. In: C. Atkinson; N. Baker; S. Uehara; M. Schader (Hrsg.): Proceedings 1999 Third International Enterprise Distributed Object Computing Conference (EDOC'99). Mannheim, S. 164-171

[FeLo02] Fettke, P.; Loos, P.: Methoden zur Wiederverwendung von Referenzmodellen – Übersicht und Taxonomie in Becker, J.; Knacksted, R. (Hrsg.): Arbeitsberichte des Instituts für Wirtschaftsinformatik, Arbeitsbericht Nr. 90, Referenzmodellierung 2002, Methoden – Modelle – Erfahrungen

[HiLe98] Hitomi, A.; Le, D.: Endeavors and component reuse in web-driven process workflow, in California Software Symposium, Irvine, CA, USA, October, 1998

[ITIL] www.itil.co.uk

[ITSMF] www.itsmf.org

[JaBu96] Jablonski, S.; Bußler C.: Workflow Management - Modeling Concepts, Architecture and Implementation. London 1996

[JNFO00] Jennings, N. R.; Norman, T.; Faratin, P.; O'Brien, P.; Odgers B., and Alty, J.. Implementing a Business Process Management System using ADEPT: A Real-World Case Study. Journal of Applied Artificial Intelligence, 14(5):421 – 465, 2000.

[Kram99] Krampe, D.: Wiederverwendung von Informationssystementwürfen – Ein fallbasiertes werkzeuggestütztes Ablaufmodell. Wiesbaden 1999.

[Lang97] K- Lang: Gestaltung von Geschäftsprozessen mit Referenzprozeßbausteinen, DUV-Verlag, Gabler, Wiesbaden, 1997

[McIl69] M. D. McIlroy: Mass Produced Software Components. In Software Engineering, ed. P. Naur, B. Randell: NATO Science Committee, Januar 1969, S. 138 — 150.

[Over03] Overhage, S.: Towards a Standardized Specification Framework for Component Development, Discovery, and Configuration. In: Bosch, J., Szyperski, C., Weck, W. (Eds.): Proceedings of the Eighth International Workshop on Component-Oriented Programming (WCOP 2003).

[Schm03] Schmidt, R.: Web Services Based Architectures to Support Dynamic Inter-organizational Business Processes, pp. 123 – 136, Proceedings Web Services - ICWS-Europe 2003, Editors: M. Jeckle, : Zhang

[Schm04] Schmidt, R.: IT-Service-Management – State and Perspectives. 4. itSMF Kongress Hamburg 2004

[Schu01] Schulze, D.: Grundlagen der wissensbasierten Konstruktion von Modellen betrieblicher Systeme. Aachen 2001

[Szyp97] C. Szyperski: Component Software - A Market on the Verge of Success. The Oberon Tribune, 2(1), 1997.

[Szyp98] C. Szyperski: Component Programming, Beyond Object-Oriented Pro-gramming. Addison Wesley, New York, 1998.

[WLMP04] Wroe, C.; Lord, P.; Miles, S.; Papay, J.; Moreau, L.; Goble, C.: Recycling Services and Workflows through Discovery and Reuse. http://www.allhands.org.uk/2004/proceedings/papers/218.pdf

[Wolf01] Wolf, S.: Wissenschaftstheoretische und fachmethodische Grundlagen der Konstruktion von generischen Referenzmodellen betrieblicher Systeme. Aachen, 2001

Comparing the Impact of Service-Oriented and Object-Oriented Paradigms on the Structural Properties of Software

Mikhail Perepletchikov, Caspar Ryan, and Keith Frampton

RMIT University, School of Computer Science and Informational Technology
{mikhailp, caspar, keithf}@cs.rmit.edu.au

Abstract. Service-Oriented Architecture (SOA) is a promising approach for developing enterprise applications. While the concept of SOA has been described in research and industry literature, the techniques for determining optimal granularity of services and encapsulating business logic in software are unclear. This paper explores this problem using a case study developed with two contrasting approaches to building enterprise applications that utilise services, where one of the approaches employs coarse-grained services developed based on the principles of Object-Orientation (OO), and another approach is based on embedding business rules and logic into executable BPEL scripts and constructing a system as a set of fine-grained services. The quantitative comparison based on a set of mature software engineering metrics showed that a system developed using the BPEL-based approach has a potentially higher structural complexity, but at the same time lower coupling between software modules compared to an OO approach. It was also shown that some of the existing software metrics are inapplicable to SOA, hence new metrics need to be developed.

1 Introduction

Service-Oriented Architecture (SOA) is an approach for constructing integrated enterprise software systems that employ services, where a service represents a function that is self-contained, and does not depend on the context or state of other services [7]. SOA-based systems are defined as a collection of interacting services that offer well-defined interfaces to their potential users. One of the driving factors behind SOA is its business alignment [2, 17]. Businesses depend on information technology for their everyday tasks, and as such, the logic and rules that drive the business are integral part of software. The traditional approach is to code business logic into software itself, whereas SOA in conjunction with Business Process Modelling (BPM) allows situating business logic within executable business processes that can be designed and implemented by business modelers with the aid of tool support, thus providing a higher level of abstraction for encapsulating business logic, and facilitating reconfiguration.

Although various publications have described benefits of embedding business logic into executable business processes, including increased maintainability and reusability of software [5, 14], such descriptions have not been empirically evaluated or supported by case studies. Therefore, this paper provides an initial quantitative evaluation of the impact of embedding business logic into business processes on the structural software attributes of size, complexity, coupling and cohesion.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 431 – 441, 2005.
© Springer-Verlag Berlin Heidelberg 2005

The contribution of the paper is as follows. A case study was used to illustrate issues related to granularity of services in service-oriented development in order to drive further research in the area of service granularity and implementation of business logic in software with the aim of specifying guidelines applicable to SOA development. To facilitate this case study, a prototypical enterprise application was designed and implemented using two contrasting approaches. The approaches represent two extremes in service granularity, where one of the approaches is based on the principles of Object-Orientation (OO), consisting of one coarse-grained service with business logic embedded into a hierarchical OO design structure, and another approach based on embedding business rules and logic into executable BPEL4WS scripts and constructing a system as a set of fine-grained services. By investigating two extremes in service granularity this paper explores issues related to coarse and fine grained services with the aim of finding a right balance between the extremes in granularity since in practice neither of them is ideal. Note that a fine grained OO-based design and a coarse grained BPEL-based design are also possible in theory, and as such, may be investigated in future work.

The impact of these approaches on the structural software attributes of size, complexity, coupling, and cohesion was quantitatively measured by applying a set of eight established software engineering metrics to the resulting designs and prototypical implementations. Thus, the applicability of some of the existing metrics to SOA was indirectly evaluated. Finally, the initial systems were extended and measured in order to evaluate changes in both approaches using the same metrics.

The results show that systems developed using the OO-based approach exhibit higher inter-module coupling, but at the same time have potentially lower structural complexity. Furthermore, there is a need for new metrics specifically tailored to SOA since the existing metrics were not immediately applicable to the BPEL-based approach. Such metrics will be derived and evaluated in future work.

The rest of the paper is organised as follows: Section 2 presents background material including discussion of the concepts of SOA and BPM, and a short description of the software attributes under investigation. Section 3 provides a detailed description of the case study including methodology, metrics used to quantify the designs/implementations, and illustration of the actual designs. The metrics based evaluation of the case study is presented and analysed in Section 4. Finally, Section 5 closes with conclusions and a discussion of future work.

2 Background

This section serves two purposes. Firstly, it briefly describes key concepts of Service-Oriented Architecture (SOA) and Business Process Modelling (BPM). Secondly, it presents the structural software attributes being investigated.

2.1 Characteristics of SOA

SOA is an abstract concept of how software services should be composed and orchestrated. A conceptual model of SOA consists of two primary parties: a service provider, who publishes a service description and realises the service; and a service consumer, who finds the service description in a registry and invokes the service [2].

The notion of a service is similar to that of a component, in that services, much like components, are independent building blocks that collectively represent an application. However, services are coarser grained than components; and they should exhibit complete autonomy from other services, meaning that each service should be implemented separately from other services resulting in a loosely coupled system [7].

For the purpose of this paper, SOA is defined as a *software development paradigm based on the concept of encapsulating application logic within independent, loosely coupled, stateless services that interact via messages using standard communication protocols, and can be orchestrated using business process languages*. This particular definition was chosen since it captures the main essence of SOA from both, representational (architectural) and development perspectives.

In SOA, instead of thinking of services as interfaces to software functionality that connect to other interfaces, organisations should consider services as enablers of business processes. Technically, a business process in SOA is a service; hence developers can compose local or external services into executable business processes that are exposed externally as services.

2.2 Business Process Modelling

Business processes reflect workflows within and between organisations. Business process modelling (BPM) describes activities that interact with various intra/inter-organisational elements while supporting the operation of the business.

There are a large number of techniques proposed for business process modelling ranging from flow charts and workflow languages to UML and Petri Nets, each having various supporting business process languages. Until recently, Microsoft used the Pi-Calcius model with XLANG (http://msdn.microsoft.com/library/en-us/biztalks/), IBM used Petri Nets with Web Services Workflow Language (WSFL) (http://www-306.ibm.com/software/solutions/webservices/pdf/WSFL.pdf), and BPMI.org developed the Business Process Modelling Language (BPML) (http://www.bpmi.org/_vti_bin/shtml.exe/bpml-spec.htm). Such languages allow business process models to be designed and directly executed via middleware support.

The Business Process Execution Language for Web Services (BPEL) [1] is the latest in the series of BPM languages, uniting the ideas from the XLANG and WSFL specifications. It is arguably the most widely used language since it was developed by a consortium of major software vendors (such as IBM, Microsoft, and BEA). BPEL is currently in the standardisation stage at OASIS (http://www.oasis-open.org/), having adopted a model that is similar to the one promoted by BPML, thus facilitating a convergence of standards in the industry [14].

2.3 Structural Software Attributes

A software attribute of a product is any feature or property of the product. The attributes used in this paper are structural design attributes of size, complexity, coupling and cohesion. In line with its common usage [6, 8], *coupling* is defined as a measure of the extent to which interdependencies exist between software modules[1]. *Cohesion* is defined as the extent to which elements of a module contribute to one and only one

[1] A *module* is any software entity including a class, service, or business process model.

task. *Complexity* is defined in terms of the internal work performed by a service. Finally, *size* refers to the size of a software product in terms of lines of code.

In general, low coupling and complexity, and high cohesion are desired; however coupling and cohesion are conflicting requirements that usually require compromise. Structural attributes of service-oriented software (SO) have a causal impact on external quality attributes as described by Perepletchikov et al. [15].

3 Case Study

The case study described in this section and evaluated in Section 4 investigates the impact of service-oriented and object-oriented approaches to SOA development on the structural software attributes of size, complexity, coupling and cohesion. For the purpose of the case study, a portion of an Academic Management System (AMS) was designed and implemented. Such a system is used in tertiary education institutes to administer students and staff. The business logic and rules correspond to those used at RMIT University (Melbourne, Australia). The AMS is an inter-organisational system since it communicates with external services provided by Australian Tax Office (ATO), Department of Immigration, and the University's Library[2], and thus is suitable for SOA-based case study.

The following top-level use-cases were identified during the Analysis phase: enroll student, withdraw student, allocate student to tutorial group, facilitate fee payment, obtain details, update results, change workload, and allocate staff tasks. The use-cases were ranked based on their importance. The focus of this case study is on the *enroll student* use-case, which is the most important use-case in the system according to the rankings. This use-case was designed and implemented first, thus conforming to the approach prescribed by Rational Unified Process (RUP)[12] where systems are developed iteratively in a number of increments with the most important parts of the system being developed first based on the ranking of use-cases.

For each use-case in the system a workflow was created by university administrators who have knowledge of business logic and rules required for fulfillment of each use-case. For example, the following is informal description of the *enroll student* workflow associated with the *enroll student* use-case:

- The student submits an enrollment application specifying the course he/she wishes to enroll into.
- An administrator checks that student does not have any outstanding loans with the library.
- If there are no outstanding loans, the administrator checks a number of constraints (business rules) in order to determine whether this student should be allowed to enroll into the desired course. The constraints vary based on the student type, where a student can be either International (INT) or Local (LOC), and Undergraduate (UG) or Postgraduate (PG). Each combination of such types has different business rules associated with enrollment.

[2] Example web services located on the machines other than the machine AMS was running on were used for the purpose of this case study.

- If based on the business rules the student should be allowed to enroll into the course, an administrator will request approval from the appropriate supervisor for that course.
- When the enrollment is approved, an administrator will organise a payment for the additional course depending on the student type and preferred payment method, and will finally update student data to capture the changes associated with the enrollment.
- The student gets notified about the outcome of enrollment application, the workflow ends.

3.1 Methodology

The evaluation of the effect of service granularity on structural software attributes was performed in three stages: i) firstly, *enroll student* workflow was partially[3] developed using two contrasting approaches, and a set of software metrics (the metrics are described in the following sub-section) was applied to the resulting designs and implementations in order to evaluate the impact of both approaches on the structural software attributes of size, complexity, coupling, and cohesion; ii) next, the original business rules used in the *enroll student* workflow were modified. The modifications were propagated to the designs and implementations, and the modified systems were re-measured using the same metrics; iii) finally, another use-case (*withdraw student*) was developed and integrated into the existing system. Again, the modified systems were re-measured using metrics. Conducting the evaluation in three stages allowed establishing trends related to system changes.

The first development approach was based on the principles of OO [11], consisting of one coarse-grained *enroll student* service with business logic embedded into the actual software implementation. From this point onward, this approach will be referred to as the *Object-Oriented (OO) approach*.

The second approach involved embedding business rules and logic into '*enroll student'* process, and constructing a system as a set of fine-grained services. In this approach, the actual implementation was done mainly in the process itself. This approach will be referred to as the *Service-Oriented (SO) approach*.

3.2 Metrics

Since specific metrics for measuring software attributes of SOA-based systems are yet to be defined, one of the objectives of this work was to assess the applicability of conventional software engineering metrics to SOA. A set of eight well-established metrics was chosen based on their applicability to both the OO and the procedural BPEL approaches. The decision was made to use a set of OO-specific CK (Chidamber-Kemerer) metrics [4] together with the traditional McCabes Cyclomatic Complexity (CC) [13] and Lines of Code (LOC) [8] metrics. The CK metrics consist of Depth of Inheritance Tree (DIT), Number of Children (NOC), Coupling Between Objects (CBO), Response For a Class (RFC), Weighted Methods per Class (WMC) and Lack

[3] Systems are not fully operational due to the absence of a data persistence layer, although the designs and implementations are structurally complete.

of Cohesion of Methods (LCOM). These metrics were chosen since they are well-established and highly-referenced in the research literature [3, 4, 8, 10, 16].

For the purpose of this study the metrics usage is as follows: i) the LOC measure was used to compare the *size*; ii) the CC, DIT, NOC, and WMC metrics were used to compare the *complexity*; iii) the CBO and RFC metrics were used to compare the *coupling* between classes in the OO approach, and coupling between business processes and services in the SO approach; and iv) the LCOM metric was used to compare module *cohesion* of classes, services, and business processes.

3.3 Base Case

Figures 1 and 2[4] show the static and dynamic aspects of the design developed using the *OO approach*. This design makes appropriate usage of OO constructs such as inheritance, aggregation, and association. Given that the problem of enrolling students is hierarchical by its nature since there are a number of different types of students, each with their own prerequisites and constraints; OO provides a well-documented approach for this problem [9, 11]. Such an approach is based on specialisation, and polymorphic deferral of the enrollment procedure to a particular subclass of a student during a run time, with business logic and rules defined in the `enroll()` method of a particular student type. The Java programming language was used for implementation, and the interface (wsdl file) was generated using Apache Axis 2.0 technology (http://ws.apache.org/axis2/).

The business process model representing the *enroll student* workflow is a core of the *SO approach*, where a process itself explicitly describes all the business logic and rules as shown in Figure 3[4]. The services invoked from this process are finer grained than OO counterparts performing basic operations based on data manipulation. BPEL was chosen as a modelling/execution language for the *'enroll student'* process since it is most-widely used business process language as was described earlier. For design, implementation, and deployment purposes, the Oracle BPEL Process Manager 2.1 and BPEL Designer 1.0 (http://www.oracle.com/technology/products/ias/bpel/index.html) tools were used. The services were implemented using Java programming language.

3.4 Modifying Business Logic

After the initial systems were measured, the business rules were modified to measure changes required when altering business logic. Originally, prior to enrolling student into a course, a system had to check whether the addition of a course is within the allowed load limits based on the combination of two student roles, career level (UG or PG) and residency status (LOC or INT). The modifications introduced an additional role to indicate whether the student is Part-Time (PT) or Full-Time (FT), with the constraint that INT students must be FT according to Federal Government regulations.

[4] The diagrams related to the case study can be found in the online appendix at http://www.cs.rmit.edu.au/~mikhailp/research/MIOS/appendix.pdf, which is also available from the authors on request.

Implementing these changes using the *OO approach* involved adding two interfaces for each new type of the student (`PTStudentInterface` and `FTStudentInterface`); and sub-classing `LocalUGStudent` and `LocalPGStudent` classes resulting in four extra classes as shown in Figure 1 in the online appendix[4] (shaded out section). To modify the BPEL version, the actual process was changed in order to facilitate the modified requirements. Due to the space limitations the resulting BPEL process is not shown in this paper, it can be found in Figure 4[4]. The impact of the changes on both designs in terms of structural attributes is quantified in Section 4.

3.5 Adding Additional Functionality

To determine the changes required when adding extra functionality to the system, the *withdraw student* use-case was designed and integrated using both approaches.

The implementation of this use-case in the *OO approach* required only the addition of the `withdraw()` method to the `StudentMangementService` and `StudentInterface` as shown in Figure 1[4] (circled out) and implementation of this method in the appropriate student sub-classes. The *SO approach* required developing a new '*withdraw student*' business process, as well as the addition of new operations to the services. The developed workflow is shown in Figure 5[4].

4 Comparison of the Development Approaches

The measures collected from the designs and implementations of partial AMS system are shown in Table 1. The measures were collected in three stages: after the original *enroll student* use-case was developed; after the *enroll student* use-case was modified; and after the addition of *withdraw student* use-case.

Table 1. Metrics collected by measuring designs/implementations of partial AMS

	Size	**Complexity**				**Coupling**		**Cohesion**
	LOC	***CC***	***WMC***	***DIT***	***NOC***	***CBO***	***RFC***	***LCOM***
Enroll Student - *Original*								
OO	950	16	60	4	4	14	30	-
SO	990	14	16	1	1	8	10	-
Enroll Student - *Modified*								
OO	1280	28	62	5	6	20	**36**	-
SO	1060	***19***	18	1	1	8	12	-
Enroll Student (*Modified*) and Withdraw Student								
OO	1410	30	65	5	6	20	**70**	-
SO	1440	***30***	20	1	1	16	22	-

4.1 Metrics Collection Process

Since the CK metrics were designed specifically for object-oriented systems, they are not immediately applicable to the *SO approach* due to a lack of inheritance and

aggregation. Therefore, the following assumption was made to facilitate measurement of BPEL processes and services using CK metrics:

Business Process (BP) = Service = Class

As such, the following procedures were used to measure software attributes of designs and implementations developed using both approaches:

- **Lines of Code (LOC):** i) in the *OO approach,* all java source files (excluding comments) were counted as code; ii) in *SO approach*, bpel scripts and the actual services (implemented in Java) were counted as code.
- **Cyclomatic Complexity (CC):** i) in the *OO approach*, all conditional statements and loops within implementations of method bodies were counted in order to derive CC according to [13]; ii) in the *SO approach,* all conditional statements and loops within *BP* and individual services were counted in order to derive CC again according to [13].
- **Weighted Methods per Class (WMC):** i) The WMC can be measured by either counting methods implemented within a class, or finding total CC of the methods [16]. Since the CC was already measured, the total number of methods in the system was calculated to indicate WMC in the *OO approach,* and the total number of methods in service implementations indicates WMC in the *SO approach*.
- **Depth of Inheritance Tree (DIT) and Number of Children (NOC):** i) in the *OO approach*, DIT and NOC were calculated according to [4]; ii) in the *SO approach,* DIT and NOC should always have a count of one due to an absence of inheritance in SOA. Therefore, the observation can be made that inheritance-related metrics are inappropriate for measuring SO approach since the concepts of SOA do not directly map to the concepts of OO.
- **Coupling Between Objects (CBO):** i) In the *OO approach*, the inheritance was taken into consideration when measuring CBO according to [3], if a method call is polymorphic all the classes to which the call can go are included in the coupled count. The decision was made to measure the classes included in the *enroll/withdraw student* sequence diagrams only, and then select the class with the highest CBO count for the comparison purposes (the `Student` class was selected). This decision was based on the fact that it would be inappropriate to calculate means or variances in CBO counts since there are only one (or two, in the extended case) *BP*s in the *SOA approach; ii)* In the *SO approach*, a coupling of business processes to services was measured, where a business process is said to be coupled to a service if one of them sends the message to the other.
- **Response For a Class (RFC):** i) In the *OO approach*, the RFC count includes all methods accessible within the class hierarchy and was measured according to [3, 4]. Only the response sets for the `enrollStudent()` and `withdrawStudent()` methods in the `StudentManagementService` class were counted to allow for objective comparison since the *SO approach* was based purely on these two functionalities; ii) in the *SO approach*, RFC count indicates the number of messages sent from a given business process to the associated internal/external services.
- **Lack of Cohesion of Methods (LCOM):** i) In the *OO approach*, the systems were not fully implemented thus LCOM could not be measured; ii) It was discovered that LCOM is not appropriate for the *SO approach* since business process has only one method, and services have functional nature, therefore LCOM cannot be calculated due to a lack of service (class) variables.

Note that only the direct designs/implementations artifacts were measured; Java libraries, etc. were not used.

4.2 Discussion

Prior to measuring the resulting designs and implementations, three *informal* hypotheses were defined based on a critical analysis of related literature [2, 7, 14, 17] and the authors' practical experience with SOA development:

- ***H1*:** the designs/implementations developed using *SO approach* exhibit *lower coupling* compared to those developed with *OO approach* since there are no 'strong' relationships between services and business processes in SOA-based systems.
- ***H2*:** the *SO approach* allows for *easier propagation of changes* in business logic compared to the *OO approach* since the logic is encapsulated in BPEL scripts rather than application code.
- ***H3*:** The designs/implementations developed using *OO approach* exhibit *lower complexity* than the ones developed using *SO approach* since the OO paradigm has advantages of being mature and well-established [9, 11], it can solve various problems using high-level design abstractions such as design patterns [9].

A number of general observations can be made concerning the results of the case study in regards to the above-described informal hypotheses. Firstly, the CBO and RFC counts are higher for *OO* than *SO approach*, and the count for RFC increases rapidly in OO design when a new functionality is added to the system as shown in Table 1 (highlighted in bold) showing that SOA introduces lower coupling compared to traditional approaches such as OO as was expected (***H1***).

Secondly, in *OO approach* there is a large increase in CC and LOC when business logic is modified compared to the increase in CC and LOC for *SO approach* as shown in Table 1 (underlined). As such, the observation can be made that the propagation of changes is easier in *SO approach* ***(H2)*** for this particular change.

Finally, although the measures of CC for *SO* are roughly equal to those for *OO*, there is a potentially large explosion of complexity in *SO approach* as shown in Table 1 (highlighted in bold italic). This is due to the fact that a new business process will be developed every time a new functionally is added to the system. Even though the *OO approach* has a larger count for WMC, this measure does not reflect true complexity since most of the methods counted as part of WMC in *OO approach* are accessors and mutators. Hence, a prediction can be made that CC for the fully implemented system in SO will be significantly larger than that for OO, thus supporting ***H3***.

Furthermore, in the process of obtaining measures it was noted that DIT, NOC, and LCOM metrics do not provide proper measure for attributes under investigation in the *SO approach* as explained in the previous section. Also, CC in isolation does not fully indicate complexity of a business process since it does not take into account types of BPEL constructs and interface complexities of associated services. Hence, there is a need for a set of metrics that are specifically suited for measuring structural properties of SOA-based systems. For example, by assigning particular weights to each type of BPEL activity, a complexity of business process could be measured.

4.3 Limitations

There are a number of limitations associated with the case study. Firstly, only *enroll student* and *withdraw student* use-cases were designed and implemented. Secondly, implementations are not fully operational due to the absence of a data persistence layer, although the designs and implementations are structurally complete. Such factors could influence structural attributes under investigation. Furthermore, there are many different ways of designing AMS using both *OO* and *SO approaches*, as such, not all designs will exhibit the measures presented in this paper.

Another limitation is that only functional requirements were taken into consideration when developing the case study, hence it is not clear what impact non-functional requirements, such as security and performance, could have on the structure of software and business process models. Finally, statistical analyses were not conducted, and only a small subset of software engineering metrics was used in the case study.

5 Conclusions and Future Work

This paper has demonstrated the impact of Service-Oriented and Object-Oriented development paradigms on the structural software attributes of size, complexity, coupling, and cohesion using a case study developed with two contrasting approaches representing extremes of service granularity. The resulting designs and implementations were measured in three stages: after the initial system was developed; after the business logic for the initial system was changed; and after a new functionality was added to the initial system.

The quantitative comparison based on a set of eight mature software engineering metrics suggested that: i) systems developed using the SO approach could exhibit lower coupling between software modules compare to the ones developed with the OO approach; ii) the SO approach may provide better separation between business logic and software thus requiring less modifications upon the changes to the business logic; iii) however, systems developed using the OO approach has a potentially lower structural complexity. Based on the above suggestions, a conclusion can be made that there is a need for a balance between the presented extremes in service granularity in order to maintain low coupling and complexity while allowing for easier changes to business logic and rules.

To formalise the findings presented in this paper, a set of SOA-specific metrics for measuring structural software attributes will be identified in future work since it was discovered that OO-based metrics for measuring cohesion and complexity are not readily applicable to SOA. Such metrics will be applied to the data collected from a larger case study, facilitating more formal empirical analysis and validation of described hypotheses. In addition, the issues discussed in the paper should facilitate future research into the area of service granularity and implementation of business logic in software with the aim of specifying detailed guidelines and activities applicable to SOA development.

Acknowledgement. This project is funded by the ARC (Australian Research Council), under Linkage scheme no. LP0455234.

References

[1] Andrews, T., et al., Business Process Execution Language for Web Services, Version 1.1. 2003, BEA Systems, IBM Corp., Microsoft Corp., SAP AG, Siebel Systems. http://www-128.ibm.com/developerworks/library/specification/ws-bpel/

[2] Arsanjani, A., Service-oriented modeling and architecture: how to identify, specify, and realize services for your SOA. 2004, IBM - whitepaper. ftp://www6.software.ibm.com/software/developer/library/ws-soa-design1.pdf

[3] Briand, L.C., et al. A Comprehensive Empirical Validation of Design Measures for Object- Oriented Systems. in Fifth International Software Metrics Symposium. 1998.

[4] Chidamber, S.R. and C.F. Kemerer, A Metrics Suite for Object-Oriented Design. IEEE Transactions on Software Engineering, 1994. 20(6): p. 476-493.

[5] Clune, J., BPEL in Service-Oriented Architecture, in Web Services Journal. 2005. p. 16-19.

[6] Dolado, J., A validation of the component-based method for software size estimation. IEEE Transactions on Software Engineering, 2000. 26(10): p. 1006-1021.

[7] Erl, T., Service-Oriented Architecture: a field guide to integrating XML and Web services. 2004, Upper Saddle River, NJ: Prentice Hall PTR.

[8] Fenton, N.E. and M. Neil, Software Metrics: Roadmap, in Future of Software Engineering, A. Finkelstein, Editor. 2000, ACM Press.

[9] Fowler, M., D. Rice, and D. Hoang, Patterns of enterprise application architecture. The Addison-Wesley signature series. 2003, Boston, Mass: Addison-Wesley. 533.

[10] Henderson-Sellers, B., Object-Oriented Metrics: Measures of Complexity. 1996, New Jersey, USA: Prentice-Hall.

[11] Jacobson, I., G. Booch, and J. Rumbaugh, The unified software development process. The Addison-Wesley object technology series. 1999, Reading, Mass: Addison-Wesley. 463.

[12] Kruchten, P., The Rational Unified Process: an introduction. 3 ed. 2003, Boston, MA: Addison-Wesley.

[13] McCabe, T.J. and A.H. Watson, Software Complexity. Journal of Defense Software Engineering, 1994. 7(12): p. 5-9.

[14] Pasley, J., How BPEL and SOA are changing Web services development. Internet Computing, IEEE, 2005. 9(3): p. 60-67.

[15] Perepletchikov, M., C. Ryan, and Z. Tari. The Impact of Software Development Strategies on Project and Structural Software Attributes in SOA. in Second INTEROP Network of Excellence Dissemination Workshop (INTEROP'05). 2005. Ayia Napa, Cyprus.

[16] Prnjat, O. and L. Sacks. Measuring complexity of network and service management components. in Second IEEE Latin American Network Operations and Management Symposium (LANOMS 2001). 2001. Belo Horizonte, Brazil.

[17] Singh, M.P. and M.N. Huhns, Service-Oriented Computing: Semantics, Processes, Agents. 2005, West Sussex, England: John Wiley & Sons.

The Impact of Software Development Strategies on Project and Structural Software Attributes in SOA

Mikhail Perepletchikov, Caspar Ryan, and Zahir Tari

RMIT University, School of Computer Science and Informational Technology
{mikhailp, caspar, zahirt}@cs.rmit.edu.au

Abstract. Service-Oriented Architecture (SOA) is a promising approach for developing integrated enterprise applications. Although the architectural aspects of SOA have been investigated in research and industry literature, the actual process of designing and implementing services in SOA is not well understood. The goal of this paper is to identify tasks needed for successful design and implementation of services, and investigate their effect on the project and structural software attributes in the context of SOA. This facilitates the specification of guidelines for decreasing the required development effort and capital cost of the SOA projects, and improving the structural software attributes of service implementations. The tasks are identified in the context of top-down, bottom-up and meet-in-the-middle software development strategies.

1 Introduction

Service-Oriented Architecture (SOA) is an approach for developing enterprise software systems that employ services. SOA-based systems are defined as a collection of interacting services that offer well-defined interfaces to their potential users, where a service represents a function that is self-contained, and does not depend on the context or state of other services [7].

Although the notion of a "service" is becoming increasingly popular as a means for developing large-scale distributed systems, no systematic, methodological approach to service-oriented software development exists to date [11]. Furthermore, there are conflicting opinions as to which development strategy should be used when developing SOA-based systems. These strategies include top-down, bottom-up, and meet-in-the-middle development approaches, and even though such approaches are applicable to the development of informational systems in general [1], this paper concentrates on additional constraints and properties introduced by SOA.

The contribution of this paper is as follows. Firstly, the general tasks for designing and implementing SOA-based applications were identified based on a critical analysis of related literature [2, 3, 6, 7, 13, 15-17], communication with industry practitioners and researches [16, 17], and the authors' practical experience with SOA development. Secondly, the impact of these tasks on project and structural software attributes were analytically determined. Finally, initial guidelines for improving the internal structure of services while decreasing project costs were specified.

The emphasis of this paper is on the design and implementation phases of SOA development rather than enterprise architecture or business modeling. As such, it concerns issues related to the transition from business process models to the

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 442 – 451, 2005.
© Springer-Verlag Berlin Heidelberg 2005

implementation of services in software. This lays a foundation for further study of methodological aspects covering design and implementation of SOA-based systems. In addition, the paper briefly discusses the relationship between structural software attributes and software quality attributes. Such relationship will be formalised and evaluated in future work.

The rest of the paper is organised as follows: Section 2 presents background material including important concepts of SOA, descriptions of software and project attributes under investigation, and an overview of top-down, bottom-up, and meet-in-the-middle development strategies. This facilitates identification of development tasks and their impact on project and structural software attributes in the context of SOA, and the provision of guidelines for successful design and implementation of services in SOA as described in Section 3. Finally, Section 4 closes with conclusions and a discussion of future work.

2 Important Characteristics of SOA

SOA is an abstract concept of how software services should be composed and orchestrated. A conceptual model of SOA consists of two primary parties: a service provider, who publishes a service description and realises the service; and a service consumer, who finds the service description in a registry and invokes the service [2].

The notion of a service is similar to that of a component, in that services, much like components, are independent building blocks that collectively represent an application [10]. However, services are coarser grained than components; and they exhibit complete autonomy from other services, meaning that each service is implemented separately from other services resulting in a loosely coupled system [7]. In addition, services can be composed into *composite* services or *business processes*, hence they can be reused in a context not known at the design time.

For the purpose of this paper, SOA is defined as *a software development paradigm that is based on a concept of encapsulating application logic within the independent, loosely coupled, business-aligned services that interact via messages using standard communication protocols*. This particular definition was chosen since it captures the main essence of SOA from both, representational (architectural) and development perspectives.

2.1 Software Engineering Attributes in the Context of SOA

Since the specific software engineering attributes for SOA are yet to be defined, this paper discusses how conventional software engineering attributes can be applied in the context of SOA-based design and implementation. A software attribute of a product is any feature or property of the product. The attributes used in this paper can be categorised as: project based attributes (including *capital cost* and *development effort*), and software attributes (divided into *internal structural* attributes and *external quality* attributes).

- **Project Based Attributes**

In traditional SE, the dominant part of the overall project cost is usually the development effort dictated by the estimated size of the final software product [8]. This is not

necessarily true for service-oriented development since one of the advantages of SOA is the ability to develop new applications by repurposing pre-existing services, or purchasing services from software vendors. Consequently, development effort might be low when the services are predominantly repurposed or purchased, whereas the actual capital cost can be high depending on the cost of the purchased services.

For the purpose of this paper, *capital cost* is analysed separately from *development effort*, where capital cost represents upfront project costs including: equipment, development tools, and training costs. Development effort represents ongoing costs throughout Software Development Life Cycle.

- **Internal Structural Software Attributes**

The paper investigates the impact of development strategies and their associated activities and tasks on the widely-used [4] internal structural software attributes of coupling, cohesion, and complexity.

In line with its common usage, *coupling* is defined as a measure of the extent to which interdependencies exist between implementation of services in software. *Cohesion* is defined as the extent to which elements of a service contribute to one and only one task. Finally, *complexity* is defined in terms of the internal work performed by a service. In general, low coupling and complexity, and high cohesion are desired [4].

Structural software attributes do not describe visible quality of a product, rather, they have a causal impact on external quality attributes. Identifying guidelines for decreasing complexity and coupling, and increasing cohesion of services ultimately aims to positively influence external quality attributes.

- **External Quality Attributes**

According to the quality model specified in the ISO/IEC 9126-1:2001 standard [9], there are six main external software quality attributes: functionality, reliability, efficiency, usability, maintainability, and portability.

The structural software attributes combined with various factors influence the external quality attributes, therefore a predictive model for estimating a particular quality attribute can be established in the form of:

Quality attribute = f (structural attributes, other factors)

The external quality attributes are introduced in this paper for the purpose of establishing a connection between structural properties of services and quality of SOA-based systems. The derivation of formal, measurable models for each of the external quality attribute will be described in future work.

2.2 SOA Development Strategies

There are three main strategies used for developing SOA-based enterprise applications: top-down, bottom-up, and meet-in-the-middle.

A **top-down** strategy starts with the requirements and business process models and refines them in a stepwise fashion down to a software implementation. The top-down development is often referred to as domain decomposition, which consists of the decomposition of the business domain into its functional areas and subsystems [2]. In the SOA-based top-down development, business process models provide a blueprint

for the identification of services. Services are than modeled and realised by service providers, and consumed by service consumers.

A top-down development strategy is arguably more interoperable than a bottom-up approach since avoiding language-specific types and starting with interface and message definitions can lead to a much higher likelihood of interoperability [12]. The drawback of top-down approach is that, in its full generality, it can only be applied to systems developed entirely from scratch [1].

A **bottom-up** strategy, in contrast, originates from the technical basis and tries to work upwards to the requirements and business process models by building services on a top of existing (legacy) systems. In bottom-up development, software engineers analyse and leverage APIs, transactions, and modules from legacy systems such as mainframe or ERP applications. In some cases, componentisation of the legacy systems is needed to re-modularise the existing assets to support service functionality [2]. Most distributed information systems these days involve a component of bottom-up development [1].

A bottom-up strategy includes two different activities. Firstly, developers can *add a layer of services* on top of legacy systems by creating wrappers and adaptors for legacy software. Secondly, legacy systems can be *refactored* in such a way that the external behavior of the code remains the same, whereas the internal structure becomes SO.

A **meet-in-the-middle** strategy is essentially a combination of top-down and bottom-up techniques. Currently, the techniques for meet-in-the-middle approach are not well understood. To the knowledge of the authors, the only well-described technique is a goal-service modeling proposed by Arsanjani [2].

In this technique, high-level business process functionality is externalised for coarse-grained services. Examining the existing legacy functionality and deciding how to create adaptors and wrappers allows specifying finer-grained services. Finally, a cross-sectional approach can be applied in order to reduce the number of candidate services that have already been identified. This technique also ties services to goals, performance indicators, and metrics.

3 The Impact of Development Strategies on Software Attributes

In order to facilitate investigation of the impact of development strategies on project and structural software attributes, the top-down, bottom-up, and meet-in-the-middle strategies have been divided into a number of general *activities*, where an activity contains a number of *tasks* for designing and implementing services in SOA-based applications. Tables 1-3 show the development strategies together with the associated activities and tasks, where a grouping of related tasks is shown in a separate cell within the table.

The impact of the identified tasks on the project and structural software attributes was analysed, and tasks have been grouped together based on their influence on a particular attribute under investigation. *The up (↑) and down (↓) arrows are used to indicate the impact of a particular task on the attributes under investigation. The*  *and*  *symbols are used to indicate whether such impact is positive or negative in regards to a particular attribute.*

In situations where a task influences attribute/s other than the one it was originally intended for, the impact of this task on such attribute/s is shown in brackets together with arrows indicating negative/positive influence. For example, the '*provide training'* task in the "Building services" activity of the top-down strategy directly influences the *capital cost* attribute thus having following indicator associated with it - ↑ (**Development Effort** ↓). *This states that the 'provide training' task will increase capital cost, but at the same time decrease development effort.*

Any given combination of tasks constitutes a *guideline* that can be selected based on the requirements of the project. Note that the aim of this paper is not to identify a concrete development methodology, but to investigate the impact of tasks on project and software attributes, and establish initial guidelines for SOA-based design and implementation. These guidelines are presented in the following sub-sections.

3.1 Guidelines for Top-Down Development

There are various activities involved in realising services in a top-down approach. Such activities include *building* services from scratch, *repurposing* existing services, and *purchasing* services.

The crucial task of building services from scratch is to identify the smallest units of software (service components) that can be reused in different contexts. Service components should be then composed into coarser-grained composite services or business processes. By structuring the system as a set of highly-reusable, loosely-coupled services, companies can increase Return on Investment (ROI) due to decreased maintenance costs and ability to repurpose services in future projects.

Also, organisations should purchase Enterprise Service Bus (ESB) implementations to facilitate connectivity, routing of messages, etc. In addition, Integrated Services Environments (ISE) should be used to design, configure, test, and debug business processes. Although these products might increase the capital cost of the project, they will reduce the required development effort as shown in Table 1.

To facilitate the future repurposing of services, an enterprise should incorporate a private service registry to centralise published service descriptions into one accessible resource. When repurposing services, pre-existing services should be integrated into the system using integration/composition code, the services themselves should not be modified. This will save time on testing since there is no need to conduct unit tests on the pre-existing services, only integration tests are required. Finally, prior to making a decision to purchase services, an enterprise should conduct a Cost-Benefit Analysis to evaluate pros and cons of purchasing services instead of building them in-house.

3.2 Guidelines for Bottom-Up Development

An important task in bottom-up development is to use software quality metrics to measure the structural design properties of legacy systems in order to decide whether it is best to *refactor the system*, or simply *add a layer of services* to it. In future work, the suitability of existing structural complexity measures will be evaluated, and a threshold for acceptable level of complexity will be established. Also, it is important to take business process models into account when determining required services.

Table 1. The impact of *top-down* strategy on the project and structural software attributes

Attributes / *Activities*	**Project**		**Structural Software**		
	Capital Cost ↓ (CC) ↑	**Devel. Effort** ↓ (DE) ↑	**Complexity** ↓ (C1) ↑	**Coupling** ↓ (C2) ↑	**Cohesion** ↑ (C3) ↓
Building services	- Have existing team of developers ↓ - Provide training ↑ (DE ↓) - Purchase standardized middleware and development tools (eg. ESB) ↑ (DE ↓) - Establish standard documentation/ reference models ↑ (DE↓) - Maintain private registry of services ↑ (ROI ↑)	- Build iteratively ↓ - Use mature software development processes ↓ (CC ↑) - Group development teams around logical business tasks ↓ - Build for reuse ↑ (ROI ↑ C2 ↓)	- Apply MDA approach to decompose business processes (BP) into fine-grained service components ↓ (DE ↓) - Implement service components using principles of OO ↓ - Decompose highly-complex components ↓ - Encapsulate global data in a dedicated service ↓	- Identify the smallest units of software that can be reused in different contexts (service components) ↓ - Couple service components and services through interfaces only, not through implementation ↓ - Specify simple, concise interfaces ↓ - Avoid embedding workflow aspects within services implementation ↓	- Develop fine-grained service components ↑ - Compose service components into composite services only if resulting service represents a concrete business function ↑ - Avoid embedding application policies such as security, SLAs, and QoS within services themselves ↑
Repurposing services	- Hire a business modeling expert to identify existing services that can be reused in new application ↑ (DE ↓) - Utilise existing middleware and development tools ↓	- Reuse preexisting services identified from private registry ↓ - Embed composition code necessary to support new capabilities into BPs, not in individual services ↓	N/A (the internal structure of services remains intact)	N/A (the internal structure of services remains intact)	N/A (the internal structure of services remains intact)
Purchasing services	- Purchase services from known vendors ↑ (DE ↓) - Perform adequate Cost-Benefit Analysis to evaluate the costs and benefits of developing /purchasing services ↑(DE↓)	- Develop only specific services, purchase the rest ↓ - Purchase fine-grained services, but build coarse services in-house ↓ - Repurpose if possible ↓	N/A (we cannot influence the internal structure of purchased services)	N/A (we cannot influence the internal structure of purchased services)	N/A (we cannot influence the internal structure of purchased services)

When refactoring legacy systems, it is advisable to start small, focusing on strongly-coupled and highly-complex modules. This will allow measuring ROI before making a large commitment, and gain experience before taking on larger problems. To reduce development cost when refactoring existing systems, an organization should make an effort to employ people who were involved in the architecture, design, and implementation of such systems as shown in Table 2. To reduce development cost when adding a layer of services to legacy systems, companies should consider purchasing commercial off-the-shelf software service adaptors/wrappers. In addition, the existing resources should be utilised as much as possible.

The main factor influencing the internal structural properties of services in bottom-up development is the granularity of services. Developers should make an effort to develop fine-grained services, consequently increasing cohesion, and decreasing complexity and coupling.

3.3 Guidelines for Meet-in-the-Middle Development

The bottom-up approach can lead to poor business-service abstractions since the design is usually dictated by the existing IT environment, rather than business needs. On the other hand, a top-down strategy might cause insufficient, non-functional requirement characteristics, and provide an impedance mismatch on the service and component layer [17]. Therefore, a meet-in-the-middle strategy is highly recommended.

The meet-in-the-middle is potentially the most expensive approach, but should result in a more-complete set of business-aligned services, consequently increasing ROI as shown in Table 3. The tasks for improving structural software properties in a meet-in-the-middle development include a combination of previously-described guidelines for top-down and bottom-up software development strategies.

3.4 Conflicting Factors

There are a number of conflicting factors that negatively influence some of the attributes, while contributing positively to others. Such factors introduce trade-offs between project cost and software quality, hence they should be carefully analysed by managers and software engineers in order to decide on a particular course of action.

Two major conflicting factors were identified: Firstly, the *build for reuse* task in the "Building services" activity of the top-down strategy results in higher development effort, but at the same increases ROI and improves implementation-level coupling of services as shown in Table 1. Hence, a trade-off between increased reusability and higher development cost can be observed. This is due to the fact that building a reusable unit (service) requires three to five times the effort needed to develop a unit (service) for one specific purpose [5]. On the other hand, highly-reusable services can decrease future development costs, consequently increasing ROI. Also, highly-reusable services will exhibit low coupling since they are built as totally independent software units. When building for reuse, project managers should consider these issues, so that an informed decision can be made regarding development for reuse.

Secondly, the granularity of services influences a number of attributes. For example, *developing coarse-grained services* when adding a layer of services to legacy systems will decrease the development efforts since it is easier for developers to

Table 2. The impact of *bottom-up* strategy on the project and structural software attributes

Attributes / *Activities*	**Project**		**Structural Software**		
	Capital Cost ↓ (CC) ↑	**Devel. Effort** ↓ (DE) ↑	**Complexity** ↓ (C1) ↑	**Coupling** ↓(C2)↑	**Cohesion** ↑(C3)↓
Refactoring legacy systems	- Employ people who were involved in the architecture /design of legacy systems ↑ (DE ↓) - Purchase utility (general-purpose) services ↑ (DE ↓) - Maximise use of existing resources (eg. DBs) ↓	- Refactor iteratively ↓ - Focus on strongly-coupled and highly complex modules ↓ (C1 ↓ C2 ↓ C3 ↑) - Purchase service adapters for modules that are loosely-coupled and highly cohesive (no refactoring needed) ↓ (CC ↑)	- Share complexity across refactored service components ↓	- Remove implementation coupling by ensuring that refactored modules and modules with service adaptors communicate strictly through the interfaces ↓	- Refactor existing modules into fine-grained service components ↑
Adding a layer of services to legacy systems	- Employ people who were involved in the architecture /design of legacy systems ↑ (DE ↓) - Use COTS service adaptors ↑ (DE ↓) - Maximise use of existing resources ↓	- Develop coarse-grained services ↓ (C1 ↑ C2 ↑ C3 ↓) - Establish ESB and incrementally add services to it ↓ - Remove dependencies between systems that share infrastructure ↑ (C1 ↓ C2↓)	- Legacy systems should interact only through service layer ↓	- Avoid combining functionality from different legacy systems into one service ↓	- Add fine-grained services ↑

Table 3. The impact of *meet-in-the-middle* strategy on the software attributes

Attributes / *Activities*	**Project**		**Structural Software**		
	Capital Cost ↓ (CC) ↑	**Devel. Effort** ↓ (DE) ↑	**Complexity** ↓ (C1) ↑	**Coupling** ↓(C2)↑	**Cohesion** ↑(C3)↓
Adding a layer of services to legacy systems	- Employ people who were involved in the architecture/ design of legacy systems ↑ (DE ↓) - Establish standard documentation models ↑ (DE ↓) - Maximise use of existing resources ↓	- Examine legacy systems to determine services that can be developed by externalising existing functionality ↓ - Apply cross-sectional approach [2] to cut down the number of candidate services ↓	- Combination of top-down and bottom-up approaches	- Combination of top-down and bottom-up approaches	- Combination of top-down and bottom-up approaches

generailise existing functionality into coarse-grained service interfaces. Also, coarse-grained services can improve network performance since they require less communication than fine-grained services. On the other hand, creating coarse-grained services introduces increased coupling and decreased cohesion [14], resulting in lower system quality in terms of maintainability, reliability, and efficiency. Therefore, project managers should make a trade-off in regards to expected granularity of services based on the particular project constraints.

4 Conclusions and Future Work

This paper has identified general tasks for the design and implementation phases of SOA-based development in the context of top-down, bottom-up, and meet-in-the-middle strategies. The impact of such tasks on project and structural software attributes has been qualitatively analysed. The tasks were combined into general *guidelines* for improving the internal structure of SOA-based software, and decreasing capital cost and development effort. Although the guidelines presented in this paper have not been empirically evaluated, they could be used by project managers and software engineers in order to determine a suitable development approach given particular quality requirements, project constraints, and application types.

To formalise findings presented in this paper, a suite of SOA-oriented metrics for measuring and quantifying project and software quality attributes will be identified in future work. Such metrics will be applied to the data collected from available SOA-based projects, consequently facilitating an empirical evaluation of the presented guidelines.

In addition, the issues discussed in the paper should facilitate future research into design and implementation of services in SOA. For example, the paper described two of the main issues related to SOA-based development that need to be investigated in future work: i) can services be made sufficiently independent so as to be reused in entirely different applications, whilst minimising development effort?; and ii) what is the optimal granularity of services?

Finally, the recommendations for *directly* influencing external quality attributes during the development will be provided in future work. For example, to increase *efficiency*, an organisation could develop/purchase service-oriented messaging backbone to communicate in formats other than XML since XML parsing and manipulation are very resource consuming.

Acknowledgement. This project is funded by the ARC (Australian Research Council), under Linkage scheme no. LP0455234.

References

[1] Alonso, G., et al., Web Services: Concepts, Architectures and Applications. 2004, Heidelberg, Germany: Springer-Verlag.

[2] Arsanjani, A., Service-oriented modeling and architecture: how to identify, specify, and realize services for your SOA. 2004, IBM - whitepaper. ftp://www6.software.ibm.com/software/developer/library/ws-soa-design1.pdf

[3] Barry, D.K., Web services and service-oriented architectures: the savvy manager's guide. 2003, San Francisco, CA: Morgan Kaufmann; Elsevier Science.
[4] Briand, L.C. and J. Wust, Modeling development effort in object-oriented systems using design properties. IEEE Transactions on Software Engineering, 2001. 27(11): p. 963-986.
[5] Crnkovic, I. Component-based Software Engineering. in 25th International Conference on Information Technology Interfaces. 2003. Cavtat, Croatia.
[6] Endrei, M., et al., Patterns: Service-Oriented Architecture and Web Services. 2004: IBM Redbooks.
[7] Erl, T., Service-Oriented Architecture: a field guide to integrating XML and Web services. 2004, Upper Saddle River, NJ: Prentice Hall PTR.
[8] Fenton, N.E. and M. Neil, Software Metrics: Roadmap, in Future of Software Engineering, A. Finkelstein, Editor. 2000, ACM Press.
[9] ISO/IEC, 9126-1:2001 Software Engineering: Product quality - Quality model. 2001.
[10] Kotonya, G., et al. A service model for component-based development. in 30th EUROMICRO Conference. 2004. Rennes, France.
[11] Kruger, I.H. and R. Mathew. Systematic development and exploration of service-oriented software architectures. in Fourth Working IEEE/IFIP Conference on Software Architecture. 2004. Oslo, Norway.
[12] Lehmann, M., Deploying large-scale interoperable Web Services infrastructures, in Web Services Journal. 2005. p. 10-15.
[13] Papazoglou, M.P. Service-Oriented Computing: concepts, characteristics and directions. in International Conference on Web Information Systems Engineering. 2003. Roma, Italy.
[14] Perepletchikov, M., C. Ryan, and K. Frampton. Comparing the Impact of Service-Oriented and Object-Oriented Paradigms on the Structural Properties of Software. in Second International Workshop on Modeling Inter-Organizational Systems (MIOS'05). 2005. Ayia Napa, Cyprus.
[15] Singh, M.P. and M.N. Huhns, Service-Oriented Computing: Semantics, Processes, Agents. 2005, West Sussex, England: John Wiley & Sons.
[16] Yang, J., M.P. Papazoglou, and B. Orriens, Service component: a mechanism for Web Service composition reuse and specialization. Journal of Integrated Design and Process Science, 2004. 7(4): p. 1-18.
[17] Zimmermann, O., P. Krogdahl, and C. Gee, Elements of Service-Oriented Analysis and Design: an interdisciplinary modeling approach for SOA projects. 2004, IBM - white-paper. http://www-128.ibm.com/developerworks/library/ws-soad1/

An Hybrid Intermediation Architectural Approach for Integrating Cross-Organizational Services

Giannis Verginadis, Panagiotis Gouvas, and Gregoris Mentzas

Institute of Communication and Computer Systems,
National Technical University of Athens, Greece
jverg@softlab.ntua.gr, pgouv@mail.ntua.gr,
gmentzas@softlab.ntua.gr

Abstract. Nowadays, workflow research has shifted from fundamentals of workflow modelling and enactment towards improvement of the workflow modelling lifecycle and integration of workflow enactment engines with new enabling technologies for process invocation. These efforts along with the workflow component reusability trend, aim at tackling the issues concerning the dynamic and distributed environment of the e-business domain. Other totally distributed technologies like the intelligent multi Agent Systems proved to face some of the special requirements of conducting business through Internet, but they still present credibility problems concerning the overall control of the processes. We propose a web-based «intermediation hybrid architecture» for integrating services by exploiting and combining the advantages of strict centralized topologies that use workflow engines, with totally distributed systems which use agent technologies.

1 Introduction

It is true that the major problems in the cross-organizational domain are related to the dynamic and distributed nature of the Internet environment. Such an environment creates the necessity of constantly altering, modifying and updating the related business processes, a fact that makes the creation and monitoring of systems that operate with respect to these dynamic conditions a very demanding task. Consequently, the removal of a work item from a workflow and its asynchronous re-insertion can cause problems. Another major issue is related to the special characteristics of these services. Since they are distributed across physical and geographical boundaries, any solution architecture must support an equivalent degree of distribution.

We can understand that the traditional solutions which are focused simply on the provisional modelling of processes are no longer sufficient due to the fact that the traditional workflow management systems have rigid, centralized architectures which do not operate across multiple platforms.

On the other hand, employing a distributed network of autonomous software agents that can adapt to changing circumstances would result in an improved workflow management system as it was argued in the agent-based workflows in [1,2] where the software agents take full responsibility for process provisioning, enactment and compensation. Their consolidated use instead of web services is a fact based on the agents' intelligence and capability to communicate and react to stimulations.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 452 – 460, 2005.
© Springer-Verlag Berlin Heidelberg 2005

But nowadays, more fluid business processes are needed, such as in e-commerce, or e-government. This is stressed out in the scenario presented in Section 4, where we focus on cross-organizational services addressed to companies that require interaction with public administrations other than those of the country of origin. The services offered by government, national and regional administration agencies as well as commerce and industry chambers include simple informational and complex transactional services (issue of a legal document) with particular bureaucratic, disintegrated and dispersed characteristics. Those characteristics create the need for adaptive workflows flexible enough to constantly alter the responsible actor in each task (human interaction-totally automated response to a request), or even the process logic itself (e.g. task redesign in case a new law dictates the need for the acquisition of a new document for the registration of a foreign branch). In such situations, it is not always possible to predict in advance all the parameters that may be important for the overall processes.

The main contribution of our so-called «hybrid intermediation architecture» is that it upgrades the agent-enhanced cross-organizational approaches made so far by alleviating the restraints in the intelligence of the agents, using two Agent layers with one workflow engine for creating a single one-stop point for electronic services. Our purpose is to fulfill the increased needs of the cross-organizational domain.

In section 2 we refer to a brief review of similar efforts that try to tackle the special problems of the e-business cross-organisational domain. The Hybrid technological approach is described in details in section 3, where Section 4 presents a specific scenario. Section 5 concludes the paper and discusses directions of future work.

2 Review of Existing Work

The software agent architectures for decentralized workflows can be resolved into three major categories [3]: *Agent-based* workflows where agents take full responsibility for process provisioning, enactment and compensation. *Agent-enabled* workflows, where agents appear as brokers that can invoke workflow instances in different workflow geographically dispersed engines. *Agent-enhanced* workflows which are achieved by combining a layer of agents with a commercial workflow engine. The agent layer is given responsibility for both the provisioning and compensation phases of business process management, whilst the underlying WFMS handles process enactment.

In efforts like [4] agents are incorporated as formal methods for process enactment. They adopt formal approaches to the service composition, but as agent-based approach lacks of the advantages of centralized monitoring and coordination. Helal et al. [5] use 3-tiered agent architecture for workflow enactment comprising an agent-enabled approach that has the disadvantage of restraining the agents' abilities since they can only invoke distributed workflow instances acting just as brokers.

There are several research efforts that belong to the Agent-enhanced approaches that present similarities to our work like in [3]. These agents correspond to activities one by one losing their intelligent notion and resembling to pieces of ordinary software. One notable effort in this area of distributed workflows is that of Yoo et al. [6]. Their Agent-enhanced approach comprises a workflow engine which uses also blocks, with one agent layer. The main differences with our Hybrid Intermediation

Architecture is firstly that their *Maximal Sequence Model* perceives the workflow blocks only as a group of sequential workflow tasks without any semantic connection between them (in contrast to workflow blocks in [7]) and secondly the use of agent technology is constrained in only one layer of «non-intelligent» agents that just obey the workflow engine.

Efforts like Cb-business project [8], where an intermediation scheme was developed for services offered by governmental and business service providers across the European Union, are examples of totally centralized approaches orchestrated usually by a workflow engine. The exact opposite approach can be found in research efforts like in [9, 10] where totally distributed approaches to cross-organizational workflows using the agent technology for modeling, invoking and executing workflows, were argued.

It has become obvious that the totally centralized approaches proved to be not flexible enough, in cases where business process reengineering and task reconfiguration is a constant necessity. On the other hand, the totally distributed approaches of Agent technology can fulfil the special dynamic requirements of the e-business and e-government domain, but they lack of a credible overall control.

Our «Hybrid» approach use the characteristics of both Agent-enhanced and Agent-Based workflows since it combines two layers of agents (responsible for process provisioning, compensation and enactment in some cases) with a workflow engine which handles the monitoring and some of the process enactment. The Hybrid Intermediation Hub upgrades the agent-enhanced cross-organizational systems by alleviating the restraints in the intelligence of the agents (Table 1).

Table 1. Workflow Systems using Agent Technology

Systems / Characteristics	Agent-Enabled	Agent-Based	Agent-Enhanced	Hybrid Hub
Centralized control	√	-	√	√
Distributed Use	√	√	√	√
Easy monitoring	-	-	√	√
Flexible	-	√	√	√
Intelligent	-	√	-	√

3 Operational and Technical Architecture

Considering both the complex workflow-oriented interactions among the services, and the complex interactions of agents internal to architecture in a cross-organizational environment, we propose a web-based «hybrid intermediation scheme» that can integrate the services offered by any organisation or government administration agency in the context of inter-organisational processes by exploiting and combining the advantages of strict centralized topologies with totally distributed systems that use agent technologies. This is why the proposed architecture that will be described below is called a *hybrid* one since it is based on a Multi Agent System (MAS) combined with workflow management system. Our aim is to tackle the special

needs that rise in cases like the one described in Section 4 that require at the same time centralized control, distributed use, easy monitoring, flexible and intelligent workflow execution.

From the operational perspective, the *Hybrid Intermediation Hub* consists the core of the system. The Hub that contains the workflow engine and the Multi-agent System is established in the center of a star-like topology, in which any service provider can participate by establishing one more gateway for service offerings, while end-users get a single point of contact for acquiring the desired services. When a service request is posted the hub handles all complexity of initiating, coordinating, controlling and monitoring service offering processes by combining the workflow logic with the intelligence of the appropriate Agents. Theoretically the inter-organizational communication still exists between the Hub and the Service Providers or the End Users but this time the burden is alleviated due to the structured intermediation.

In this approach, end users submit requests, which are then decomposed by the workflow management logic of the Hybrid Intermediation Hub into individual service tasks with the help of the responsible WEH Agent (presented below). These tasks are forwarded to responsible Agents that undertake the identification of the appropriate service providers, forward the service request and return the results. Each one of these Agents handles an autonomous segment of the requested service; the service providers respond to the request and the responsible Agent returns the results. This operational model is based on a solid technical architecture which is depicted below in figure 1.

Each provider offers a set of services that can either be described together with associated information flows and communication channels, using appropriate online

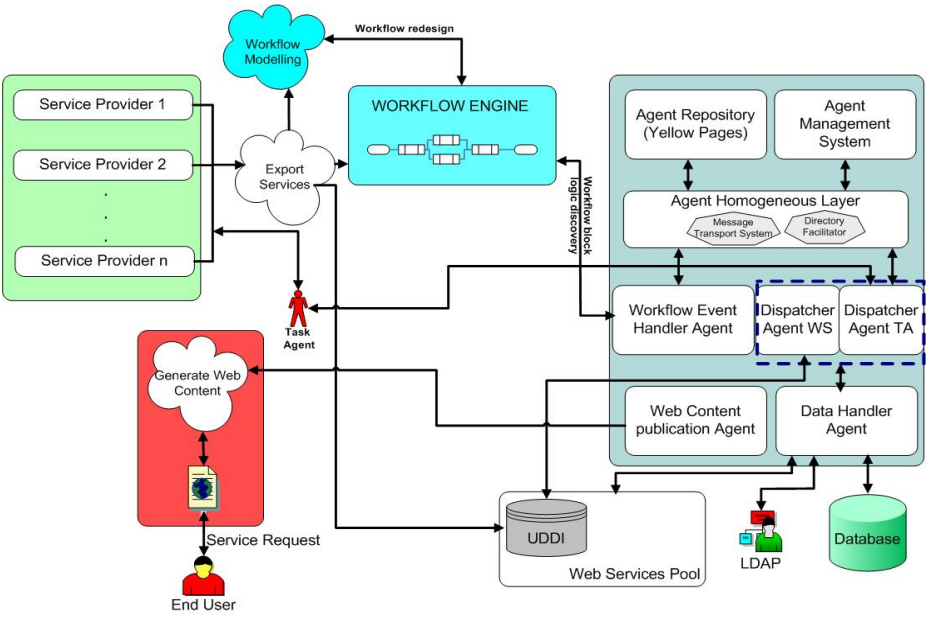


Fig. 1. Architecture

forms which then will be used by the workflow modeler as a guide to create and implement the service within our system, or can be implemented directly in case of a web service. In that way, the *Hybrid Intermediation Hub* can identify the service providers competent to serve an incoming request and employ appropriate communication channels.

In our proposed architecture we use the OpenSymphony workflow engine which is an open source Java based WFMS that can export in XML schema any information related to the description and invocation of a service. The workflow engine keeps the overall process logic which selectively passes to the responsible agents enabling in such way the centralized control and monitoring. In irreversible failure cases the engine is responsible for calling back all the agents and restarting the workflow process.

An integral part of this architecture is the Agent Platform (JADE) that facilitates the Agent-Enhanced functionality. The *Agent Repository* (Yellow Pages) and the *Agent Management System* (AMS) are functional modules of any interoperable Agent Platform. The *Agent Homogeneous Layer* consists of the Message Transport System which is also called Agent Communication Channel (ACC) and the Directory Facilitator. ACC controls the exchange of messages within the platform. The Directory Facilitator (DF) is the agent which searches the default yellow page service in the platform. In a multi-agent environment each agent is assigned specific tasks. These tasks are published through the Yellow Pages which sustains a global registry of all agents.

According to the architecture a major task that is undertaken by an agent is the *workflow block logic discovery*. It is a function that refers to the discovery of the process logic of a specific workflow block from a given log. Event handling is carried out by the Workflow *Event Handler Agent* (WEH). The agent's behavior is to process any asynchronous events that may be evoked by the run time environment of the workflow engine. For our purposes we extend the functionality of the specific Agent as it is basically the main responsible software component that is used for combining the Agent Platform with the Workflow Engine functionality by managing the exchanged XML-messages, interpreting them and invoking the competent Agents in each case. The event is categorized and re-directed to the *Dispatcher Agent WS* or *TA*. The *Dispatcher Agent WS* performs web services orchestration which is necessary for cases where the service provider side can support this kind of technology. In this case the specific Agent identifies and uses the registered web services (UDDI) according their stated function and invokes them suitably passing the necessary parameters in order to complete a certain task without human intervention. The *Dispatcher Agent TA* has the very important task of recovering from the Yellow pages and orchestrating the invocation of a number of Task Agents that each one performs a segment of the overall requested service.

The *Workflow redesign* refers to the business process reengineering which in our system becomes faster and less prone to errors due to the fact that it is made possible by altering only the specific «workflow block» that needs modification. Each time a WEH agent is invoked does not need to know if an alteration in the workflow block has happened, because anything concerning the process logic of the block that will be handled is acquired on invocation. This is a fact that makes the runtime reengineering possible in cases where the responsible for the altered block WEH agent is not yet invoked.

A dedicated Agent, the *Data Handler Agent*, is responsible for interacting with third party data sources. These sources vary from databases and LDAP servers to explicit UDDI servers. So, data handler Agent has a complex parallel behavior that manifests SQL based/LDAP queries. The UDDI interface which is separate to the Directory Facilitator registry, serves the functionality of *discovering* a web service in case that Dispatcher Agent requests for one. Finally the user's interaction is accomplished by the Web content publication Agent. This Agent is responsible for generating dynamic content. This content is a building block of a content management system such as JETSPEED (which supports Java Based Modules). In figure 2 we present the functionality of the described parts of the Hybrid Intermediation System in detail.

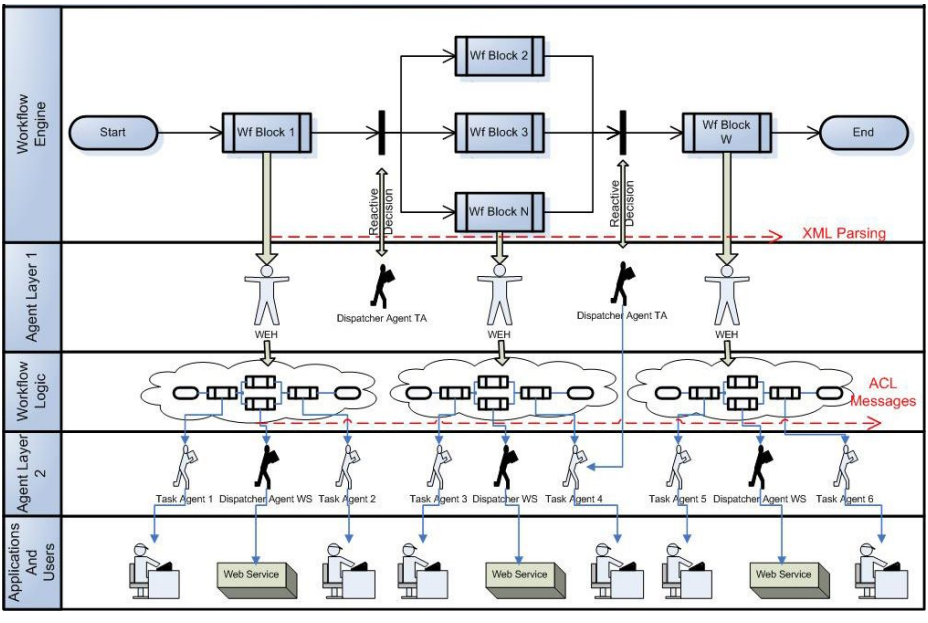


Fig. 2. Service Decomposition

In previous work [7], it was introduced the notion of workflow block as a non-trivially recurring group of consecutive (in the sense of control flow) workflow nodes, with well defined (preferably single) inputs and outputs and meaningful application-level semantics that can be isolated as an autonomous segment of a container workflow. Starting from a workflow model described in XML Process Definition Language [11] and based on such «workflow blocks», a number of Workflow Event Handler Agents are invoked to perform firstly *Workflow block logic discovery*. The Agent acquires the workflow logic of the specific block (Xml parsing), which understands and decomposes it, based on a common ontology (Process Interchange Format [12]). The decomposition is performed as the WEH Agent detects the simple tasks that are included in the «workflow block» and for each task searches and invokes the appropriate Task Agent. Secondly, they perform *Duplication,* in case a certain

block must be executed more than once with different parameters (Service Provider, Information). Finally they deliver consistent results gathered from the Task Agents.

The Task Agents are responsible for performing the subtasks that the WEH Agent instructs. The intelligence of the specific agents is not restrained although they undertake only tasks semantically simple and indivisible. They react based on strict instructions but they communicate and move autonomously in order to identify the competent Service providers (depending on their availability, their estimate response time) for fulfilling their goal. The group of the possible appropriate Service providers for a specific service is already known in the system, as they have registered their services during the workflow modeling procedure or in the UDDI registry.

The implementation is concluded using an SQL server for storing the data, an iPlanet LDAP Server and the Jetspeed open source system for posting Java Server Pages. In the next section we present a scenario upon this implementation of our proposed architecture.

4 A Scenario for the Hybrid Architecture

The overall objective of this scenario is to present the web-based, agent-enhanced «intermediation hybrid scheme» that integrates the services in the context of cross-organizational processes. The services that have been selected for our example include transactional (issuing certificates for setting up a company, as well as performing business transactions) services offered by governmental administration agencies as well as commerce and industry chambers. We present how the agent-enhanced «intermediation hybrid scheme» can facilitate such services. The workflows for providing these services, as for example the workflow for issuing a legal document were modelled within the CB-BUSINESS R&D project [8].

We demonstrate (figures 3,4) how our proposed hybrid architecture and its internal components work in case of a service that concerns the electronic acquisition of a legal document (e.g. Certificate of Origin) from a Chamber of Commerce and

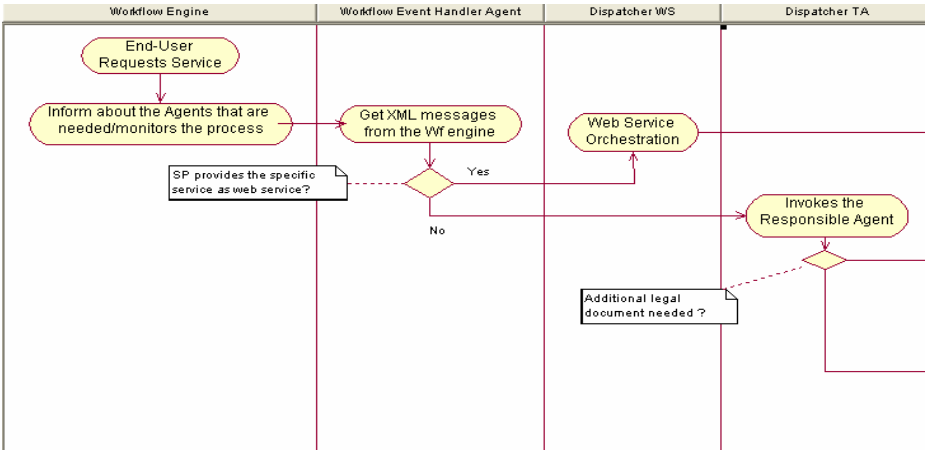


Fig. 3. Sequence Diagram 1

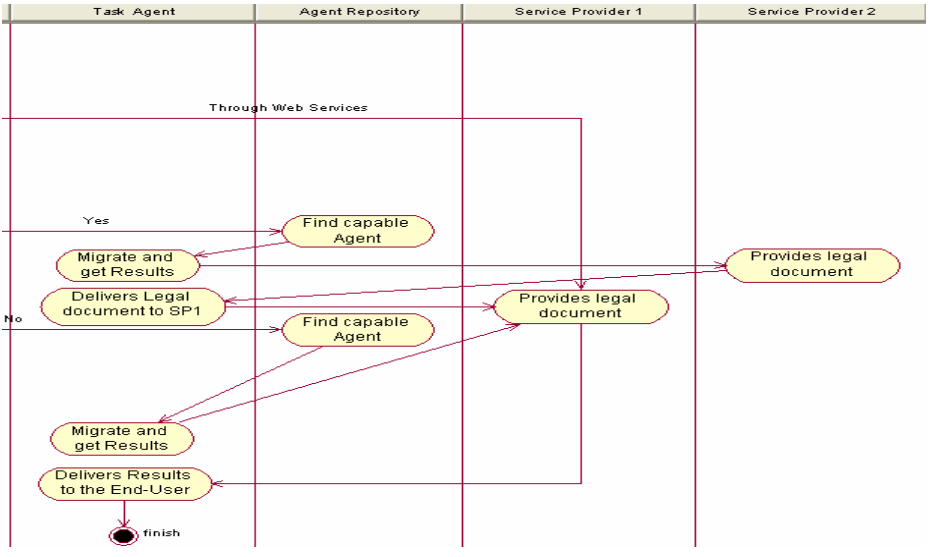


Fig. 4. Sequence Diagram 2

Industry. The businessman can request a service after he views the available service content provided by the Web Content Publication Agent. The workflow engine according to the appropriate workflow model, identifies the competent Agents that are needed and initiates, controls and monitors the process. The workflow Event Handler Agent, is responsible for interpreting the XML messages from the workflow Engine and for mobilizing the responsible Dispatcher Agent depending on whether or not the Chamber of Commerce and Industry provides the requested service as a web service. If this is the case then the Dispatcher Agent WS performs web service orchestration and delivers the legal document to the end-user. Else wise, the Dispatcher Agent TA identifies and invokes the capable agent to perform the certain task from the Agent Repository. The latter migrates to the Service Provider/s side/s in order to request the legal document providing the necessary data and at the same time any additional certificates or documents necessary. The latter documents frequently must be acquired by a different Service Provider than the one that produces the Certificate of Origin. This is undertaken by the Responsible Task Agent. Another Task Agent will attend so that the requested document reaches the end-user.

5 Conclusions

The *hybrid Intermediation architecture* that has been introduced above has a potential to achieve operational integration of cross-organizational service offerings alleviating the issues of this domain. Our effort upgrades the agent-enhanced cross-organizational approaches made so far by alleviating the restraints in the intelligence of the agents, using two Agent layers with one workflow engine for creating a single one-stop point for electronic services. Having as purpose to fulfill the increased needs of the cross-organizational domain, we have achieved the combination of a workflow

management system and an agent platform. We use Agents that can be registered or imported into a repository that basically contains reusable, easy to modify or correct, autonomous software components. These components can be located and connected accordingly. We managed to sustain central control of the workflow processes, providing easy monitoring of the workflow instances. At the same time, the distributed and flexible use was made possible by the proper integration with two layers of intelligent agents.

Additional effort is needed for the system's further validation and the extension of related work in defining and reusing workflow blocks [7] for creating a formal framework that defines recurring workflow segments in business processes. Our purpose is to implement them in Agents that can be used in our intermediation hub.

References

[1] Borghoff, U., M., Bottoni P., Mussio, P., Pareschi, R.; Reflective Agents for Adaptive Workflows. In Proceedings of the Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM97), pp 405-420, London, UK 1997.

[2] Debenham, J.; An Experimental Agent-based Workflow System. In Proceedings of the Third International Conference on the Practical Application of Agent Technology (PAAM98), pp 101-109, London, UK 1998.

[3] Yuhong, Y., Zakaria, M., Weiming, S.; Integration of workflow and Agent Technology for Business Process Management. The Sixth International Conference on CSCW in Design, Canada, July 12-14, 2001.

[4] Zeng, L.; Ngu, A.; Benatallah, B.; O'Dell, M. An agent-based approach for supporting cross-enterprise workflows. Proceedings of the 12 th Australasian Conference on Database Technologies, 123-130, Queensland, Australia, 2001.

[5] Hetal, A.; Wang, M.; Jagatheesan, A.; Krithivasan, R. Brokering Based Self Organizing E-Service Communities. Proceedings of the Fifth International Symposium on Autonomous Agents and Multi-Agent Systems 1 (1) (1998 July), pp. 7-36.

[6] Yoo, J.,J., Doheon, L., Young, H.,S., Dong, I., L.; Scalable Workflow System Model Based on Mobile Agents. Proceeding of the 4th Pacific Rim International Workshop on Multi-Agents, pp. 222-236, Taipei, Taiwan, 2001.

[7] Verginadis, G.; Gouscos, D.; Mentzas, G.; Modeling e-Government Service Workflows through Recurring Patterns. eGOV 2004, Zaragoza, September 2004.

[8] Verginadis, G.; Gouscos, D.; Legal, M.; Mentzas, G.; An Architecture for Integrating Heterogeneous Administrative Services into One-Stop e-Government. eChallenges Conference 2003, Bologna, October 2003.

[9] Blake, B., M.; Gomaa, H; Agent-oriented compositional approaches to services-based cross-organizational workflow. Decision Support Systems 2004.

[10] Jennings, N., R.; Norman, T., J.; Faratin, P.; ADEPT: An Agent-Based Approach to Business Process Management. AGM SIGMOD Record, Vol. 27, No 4, December 1998.

[11] WfMC: Workflow Process Definition Interface – XML Process Definition Language (XPDL). Technical Report WFMC-TC-1025, Workflow Management Coalition, 2001, http://www.wfmc.org/.

[12] Jintae, L., Gruninger, M., Jin, Y., Malone, T., Tate, A., Yost, G.; The PIF Process Interchange Format and Framework version 1.2. The Knowledge Engineering Review, Vol 13, No.1, pp. 91-120, March 1998.

A Framework Supporting Dynamic Workflow Interoperation

Jaeyong Shim, Myungjae Kwak, and Dongsoo Han

School of Engineering, Information and Communications University,
Yusong P.O. Box No. 77, Yusong gu, Taejon, 305-600, Korea South
{jaeyong7, mjkwak, dshan}@icu.ac.kr

Abstract. When a workflow process spans to multiple organizations, subprocess task model is an efficient way of representing remote services of other systems. The subprocess task usually represents a single service in conventional workflows. However, if a subprocess task comprises multiple services, and the number of services and the execution flow of the services cannot be decided until run-time, conventional ways of workflow design is not proper to handle such situations efficiently. All potentially reachable paths should be known at process build time in conventional workflow design. However, such an assumption does not always hold in real situations. In this paper, we propose a multi-subprocess task based framework for dynamic workflow interoperations. In the framework, we develop the multi-subprocess task model to handle a subprocess composed of multiple services that are unknown at process build time. In this paper, we also define and implement four components to support the dynamic workflow interoperation: Workflow engine, Adapter, Service Interface Repositories (SIRs), and XML messages. Adapter and SIR make a local WfMS transparent to the location and platform of the interoperating WfMSs by encapsulating external subprocesses and superprocesses. When an example scenario is implemented and evaluated in the proposed framework, the advantages of the framework are obvious in terms of automaticity, adaptability, and efficiency.

1 Introduction

As networked economy grows, business processes will be more complicated, and the capability of finding new business partners and connecting them to a business process at runtime will be essential. That is, dynamic change of trading partners will be common in future business environments [1]. Information systems in such business circumstances should have the capability to cope with changing business partners and computing environment immediately or in the middle of a process execution.

Numerous commercial WfMSs have developed their own mechanisms to interoperate with other WfMSs and enterprise applications. They usually use the concept of subflow to accommodate external services of an enterprise in B2B or e-commerce applications. In conventional workflow systems, a subprocess task (ST) usually represents just a single subprocess (SP). When multiple SPs need to be represented by a ST, conventional ways of workflow design handle it indirectly by predefining all the related STs, and connecting them in a process at process build time.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 461 – 472, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Consequently, the static specification of interoperations will be very complicated when all multiple alternative paths are specified in a process template. Furthermore, when the number and execution orders of those subprocesses are not known until runtime, and new WfMSs should be combined into running business processes, conventional ways of workflow design do not have proper means to cope with such situations. Flexibility, scalability, and adaptability should be achieved if a system is going to support interoperations. If we can make changes happening at run time transparent to a user or an application requesting a service, the WfMSs can cooperate with other WfMSs and enterprise applications more effectively.

Web services [2] can provide a standard way of communication among workflow systems. A workflow system can locate and invoke SPs, if the SPs are announced in the form of Web services. However Web services do not provide sufficient functions to support dynamic interoperations among workflow systems.

This paper proposes a framework for dynamic inter-organizational workflow interoperation to address the above requirements. A multi-subprocess task (MST) model is developed and the framework is based on the multi-subprocess task (MST) model. The MST of the framework contains multiple SPs which are unknown at process build time. The MST selects SPs one by one according to the context data of a process instance and given conditions. For the coordination of SPs mentioned above, the MST creates subprocess manager (SM) and dispatching rules. Workflow events and dynamic state transition guide the SM and dispatching rules to coordinate the SPs. Four main components are defined for the framework: Workflow engine, Adapter, Service Interface Repository (SIR), and standard XML messages.

Web Services technology can be used for the implementation of the framework, but the framework need not be confined to Web services infrastructure. We have implemented the framework using our research workflow system, ICU/COWS [3] and have studied an example scenario in the framework to evaluate the benefits of using the framework. Compared to other approaches, it shows considerable improvement in automaticity and adaptability. When there are multiple SPs which are unknown until run time, they can be easily accommodated in the framework. The framework makes adequate adaptation to the diverse changes during process execution possible without intervention of any manual operation. In addition, it requires less physical resources like time and man power.

In section 2, we discuss the requirements of dynamic workflow interoperation. Section 3 explains the multi-subprocess task model and section 4 shows the architecture of proposed framework. In section 5, an example application and evaluation of the proposed framework are discussed and we draw conclusion in section 6.

2 Requirements for Dynamic Workflow Interoperation

Figure 1 shows a business process, parts inventory management process, which spans to several organizations and internal departments. When the process starts, the first task for inventory checking is performed to get the information on required stocks. In order and inspection subprocess task (ST), if the company lacks some items in the parts, it would order those items from external suppliers or internal inspection

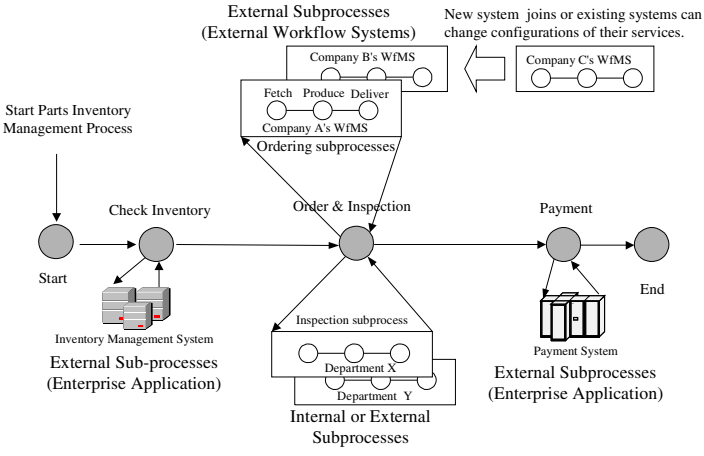


Fig. 1. An example of e-commerce process

departments. This indicates that sometimes ST has to represent multiple suppliers or internal departments. If the number of suppliers and internal departments is not known at process build time but at runtime, and if the contacting orders of them should be decided at runtime, existing approaches are difficult to cope with such situations. This is because most conventional approaches require process builders to predefine all potentially reachable paths.

If the stocks of an order are not enough, the local workflow management system orders the goods to suppliers, and then the processes in the suppliers' workflow systems are started. In this case, multiple suppliers' workflow systems may be invoked and sometimes those suppliers' workflow may invoke other suppliers' workflow. That is, if a workflow designer has to predefine all connectable subprocesses of related workflow systems in a business process template as in conventional systems, the main process may become too complex to understand. Furthermore, if the company running the process has to modify the process, it has to change the configurations of the service whenever a new system joins or some systems disappear. It is extremely inefficient.

Those problems can be addressed by introducing the notion of dynamic workflow interoperation. Dynamic workflow interoperation is accomplished by encapsulating the subprocesses, binding target subprocesses at run time, and delegating the service to the selected system based on multi- subprocess task (MST) model and multi-tiered dynamic state transition model.

3 Multi-subprocess Task (MST) Model

A MST is defined as a ST that contains multiple SPs, and in addition, it can handle the situation when the number and execution orders of them are not known until runtime. It consists of a group of subprocesses, conditions, and relevant data. A subproc-

ess pool is created at run time, and the subprocesses in the group become the members of the subprocess pool. Conditions and relevant data decide subprocesses to be invoked and the execution orders of subprocesses at runtime. For handling those unknown multiple SPs, we use rule constructor, rule dispatcher, rule interpreter, workflow events, event dispatcher, and dynamic state transition model. When the manager of a MST gets the control and receives the context data from the previous task, it generates subprocess managing instances (SMI). Rule constructor creates subprocess dispatching rules referring to the above context data and a set of given conditions. Figure 2 illustrates a schema of the parts inventory management process, where a MST represents unknown multiple SPs.

Figure 3 shows an example of constructing dispatching rules (see section 4.1) using a set of conditions and context data at runtime. Condition 1 specifies that when item A is delivered (SP A is done), department X inspects the item (SP X is invoked).

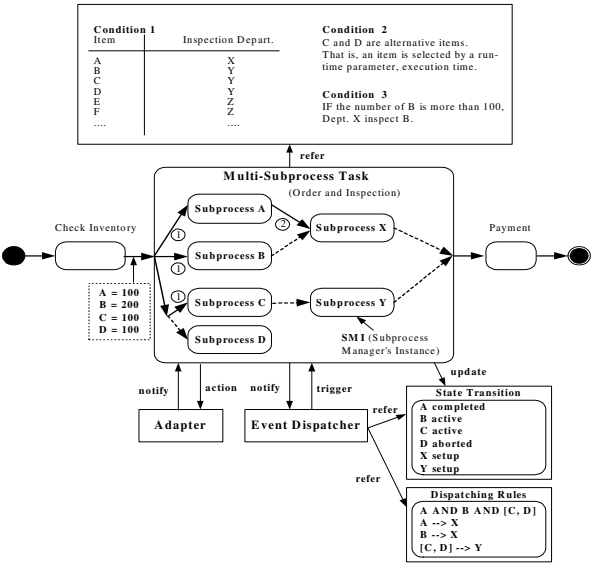


Fig. 2. An example of multi-subprocess task

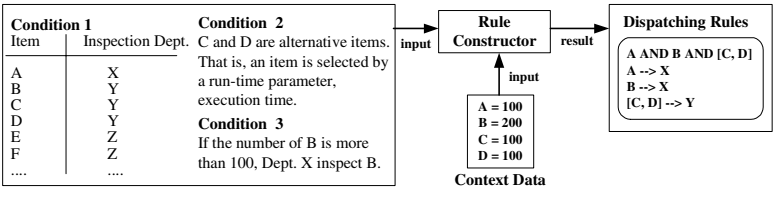


Fig. 3. Constructing dispatching rules

After the subprocess task manager (STM) finishes the setup procedure, the event dispatcher takes control. The event dispatcher organizes and coordinates the SPs according to the dispatching rules and status of the SMIs. Figure 4 shows the big picture of resolving dispatching rules. The event dispatcher invokes the rule interpreter and coordinates the SPs in the subprocess pool at runtime. The event dispatcher refers to state transition model in this process. When the event dispatcher triggers SMIs, they change their own states based on state transition model and send request messages to the systems providing the services. When the SMIs receive confirmation messages from the systems, they change their states based on state transition model, and notify them to the event dispatcher. Once the event dispatcher receives the notifications from SMIs, it triggers next SMIs using dispatching rules. In this way the event dispatcher coordinates the SPs by invoking workflow events and using dynamic state transition model. The states of SMIs change whenever the workflow events occur. The change is based on the state transition model. That is why the event dispatcher must refer to states when it determines the subsequent SP.

The dispatching rules decide both subprocesses to be executed and execution order of subprocesses at runtime. So, the rule specification for dispatching subprocess comprises subprocess identifcation and the flow structure of subprocesses. To develop constructors of dispatching rules, we extend the work of WfMC and CrossFlow's approach. We have identified two static elements and three flexible elements by referring to the WfMC's basic constructors [4] and CrossFlow's flexible elements [5]. The static elements specify the execution flow that is decided not at resolving time but at constructing time of dispatching rules. On the other hand, flexible elements of dispatching rule determine the execution flow of subprocesses at resolving time of dispatching rules. The devised 5 elements are as follows:

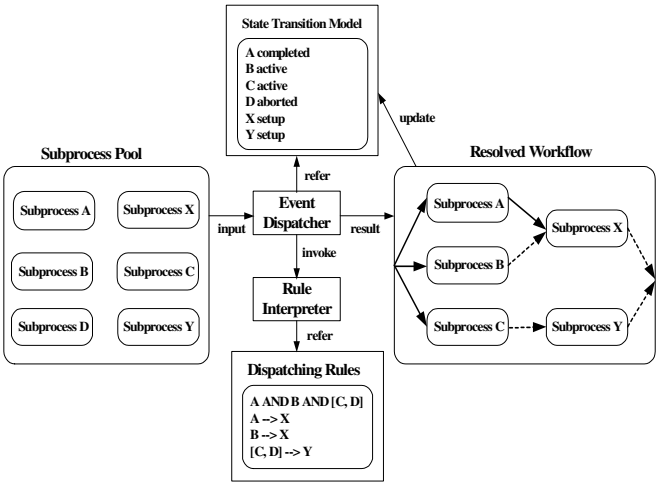


Fig. 4. Resolving dispatching rules

- Alternative SPs [A, B]: Either A or B is executed according to the run-time parameters, such as current time, execution time, and the cost of terminated activities.
- Non-vital SPs A^{nv}: This SP can be executed or omitted depending on run-time parameters.
- Optional execution order SPs {A, B}: Both A and B are execute in parallel or sequentially according to the run-time parameters.
- Static execution order SPs A → B: These SPs must be executed in sequence. The sequence is decided at constructing time of dispatching rule by a set of conditions. Run-time parameters are not considered.
- Static parallel execution SPs A AND B: A and B are executed in parallel. The sequence is decided at constructing time of dispatching rule by a set of conditions. Run-time parameters are not considered.

4 The Framework Architecture for Dynamic Workflow Interoperation

Figure 5 shows the schema of the dynamic workflow interoperation supporting framework. The framework is based on MST. As explained earlier, all the components except workflow enactment service in the schema can be implemented upon Web services infrastructure, and using Web services technology could be one of the good choices. We can expect numerous benefits by using the facilities and standards of Web services. However we do not confine the implementation of the architecture only to Web services environment because Web services still lack of many essential features to support dynamic workflow interoperaton. A more general architecture is devised, and the architecture may or may not be implemented in Web services environment.

In the framework, service providers, who want to announce their services, can register, update, and delete their service information by sending XML messages to the Global Service Interface Repository (GSIR). Whenever the service information of GSIR is changed, it broadcasts the information to the other organizations' LSIRs. If a workflow system performs a subflow task in the middle of a workflow process, the workflow engine would determine whether or not the local system could provide the service that the subflow task represents. In this light, the workflow engine should have flexible and scalable architecture for selecting and binding subprocesses at run time. Once the workflow engine finds that the local system cannot provide the service, it requests the adapter to find an appropriate external service from the external services repository pool and delegates the service to it. In this context, subprocess is transparent to users because the user or service requester can consume a service only if it requests the service with the service name and process name. The information of the location and system where the subprocess resides is not necessary. If Web services are used in this context, UDDI [6] can do the role of GSIR and LSIR instead. So GSIR and LSIR can be considered as a private UDDI in Web services environment.

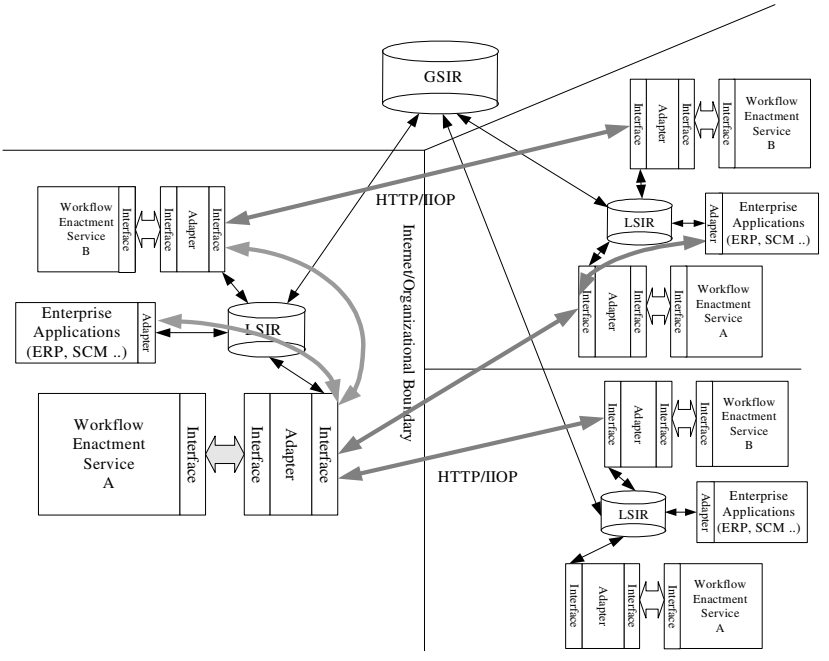


Fig. 5. Schematic view of workflow interoperation

The adapter selects an appropriate subprocess by searching LSIR with a given service name and process name. Once the serviceable workflow system is found, the adapter asks the selected system to provide the service by sending XML messages. Therefore this framework allows the service requester to use all the external subprocesses like an internal subprocess regardless of organization, location, and platform. XML messages are used to communicate with other workflow systems and enterprise applications and to access GSIR to update LSIR. Though Wf-XML messages can be transmitted over various protocols, some of them are not proper for some situations due to their characteristics. For instance, when an external subprocess is placed at a remote site, HTTP is better than other protocols because HTTP, which is a ubiquitous protocol in the Internet, is a light weight protocol than IIOP. The SMTP is not applicable because it cannot support synchronous operations. Again, the adapter and Wf-XML messages can be implemented upon Web services infrastructure. In the following subsections, we describe each module focusing on the functions of the modules.

4.1 Workflow Enactment Service

The workflow enactment service has an important role in this framework. It should be able to decide whether a service can be performed by the local system or not. Once it decides that the local system cannot provide a service, it creates a workflow manager for managing the external subprocess and delegates the service to the adapter for starting the external subprocess. The workflow engine consists of five key components [3]. Among the five classes, Workflow Requester, Global Manager, and

Local Factory are instantiated at system setup time. Workflow Manager and Task Manager are instantiated at run time responding to the requests from the instance of Workflow Requester [7]. Workflow Manager is divided into two types to handle both internal and external subprocesses. We have not designed another class to manage external subprocesses because the external subprocesses have the same functionality in the context of managing a process that consists of one or more tasks. Internal Workflow Manager manages internal processes and external Workflow Manager manages external subprocesses. Global Manager has the function of determining whether a subprocess exists inside the local system or not and the function of deciding on the types of Workflow Manager. This means that Global Manager provides location transparency to the service requester by encapsulating the subprocesses. We leave the details of the functionalities of each class to [3].

4.2 Adapter

In an object-oriented concept, delegation is a more general way of extending a class's behavior that involves one class calling another class's methods rather than inheriting them [8]. As a mediator to delegate a service from the local system to other systems, the adapter does the role of a connector to any workflow systems, as a middleware using the delegation concept. Usually the adapter object is an object that receives method calls on behalf of another object. In our framework, an adapter object sends and receives XML messages on behalf of the workflow engine. That is, the adapter encapsulates other workflow systems and dynamically binds them to the local workflow system. The Wf-XML processor within the adapter translates the programming language level method invocations to Wf-XML messages and vice versa.

The adapter has internal and external interfaces, a service layer, an XML processor, a Java Messaging Service (JMS), and a web server. Internal interfaces are defined for the workflow engine. External interfaces are defined for the external workflow systems and enterprise applications. The service layer consists of executors' instances. The executor manages request and response XML messages. Figure 6 shows the conceptual view of the adapter, web server, and workflow system. In the viewpoint of target workflow systems or enterprise applications, target workflow systems or enterprise applications can construct their own adapters. Those adapters

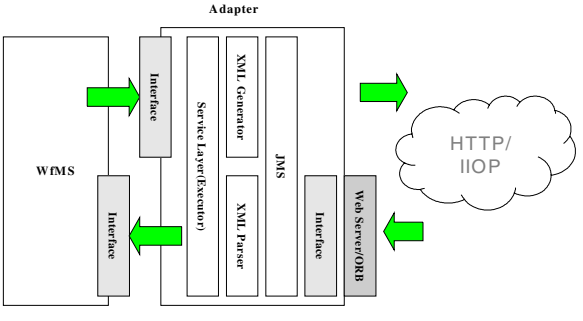


Fig. 6. A simplified architecture of adapter

must have facilities that can process standard XML messages and translate the passing data to their own data format. In Web services environment, the adapter can be implemented in the form of a proxy. On behalf of workflow system, the proxy calls remote Web services that invoke a workflow in the site, and delivers the response of the service back to workflow system.

4.3 Service Interface Repository (SIR)

SIR is a storage facility that contains service information of workflow systems. Many EDI frameworks have similar facilities that include information from other EDI systems [9]. For example, ebXML [10] and RosettaNet [11] run a service repository and master dictionary. As shown in Figure 8, to reduce the overhead of information retrieval and prevent system failure, the SIR is divided into two levels: GSIR and LSIR. Local Service Interface Manager (LSIM) manages the LSIR, which is located within the boundary of an organization. Since LSIR keeps part of service interface information in GSIR, proposed structure of SIR has advantages over that of SIRs in other frameworks. In contrast, the GSIR contains complete master information for the adapter to create service requesting XML messages. The GSIR should provide not only the available service names and process identifiers, but also the list of context data names and types for each service. For cache coherence, whenever any service interfaces are changed, either the GSIR broadcast the information or the LSIR check the updates of the GSIR. If any discrepancy between GSIR and LSIR is detected, the LSIR is updated. The GSIR is a core registry for sharing services among WfMSs. Since both the service requesting and providing systems have to access the GSIR, standardization efforts for service or process names, XML message tags, database schema, context data names and types are essential.

4.4 Wf-XML Messages

Wf-XML messages are also an important feature. By using these messages, a workflow engine can communicate with other systems via the adapter. A Wf-XML message consists of three parts, transport-specific information (WfTransport), message header (WfMessageHeader), and message body (WfMessgeBody). WfTransport is an optional section and WfMessageHeader contains information that is generically useful to all messages, such as URI (resource identifier), request or response, and so on. WfMessageBody includes operation name, request and result data [12].

Several XML messages are required to access the GSIR in our framework. They contain Document Type Definition (DTD), request and response messages, and possible exceptions. Those messages provide information to find the correct process and service. The adapter sends a request message holding service and process names. Then it receives a response message holding a process ID, process key, process description, valid state of the process, and a list of context data. The process key is a unique resource identifier for a process and the list of context data contains names and types of relevant data, which are required to create an instance of this process. With slight work, such information exchange also can be done in Web services environment. For example, Wf-XML messages should be converted into SOAP messages [13] calling remote services and returning the results.

5 The Framework Evaluation

The MST makes a main process simple and concise by dividing a complicated process into several SPs. When a MST is designed well, workflow systems and enterprise applications can automatically adapt to the changes at run time spending less physical resources. Without MST, conventional workflow systems cannot cope with the change of environments effectively. We compared our MST model with other possible approaches accomplishing the objective of coping with the changed situation at run time. Automaticity, adaptability (flexibility), business efficiency, and required physical resources are the factors in this comparison.

Figure 7 shows our approach and other possible approaches to support dynamic workflow requirements shown in Figure 1. Figure 7 (a) depicts our approach. In Figure 7 (b), (c), and (d), the shaded areas corresponds to the MST in our approach. Figure 7 (b) shows the shape of process drawn using a conventional workflow design method. All potentially reachable paths should be predefined. But this approach is extremely inefficient and almost impossible. Figure 7 (c) shows approaches that many e-procurement systems use to fetch multiple purchasing requisitions. In this approach, users manually fill out the purchasing order form through an associated application. Then the application generates multiple workflow processes prior to the process execution based on specific rules. This approach can handle multiple orders. But it cannot make the process automatically adapt to the changes at run time.
Therefore, this approach has less adaptable than our approach. Moreover, it may require more physical resources because some tasks may be performed repeatedly and

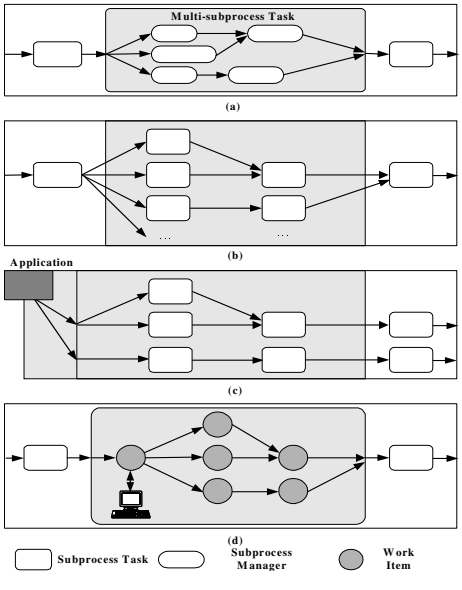


Fig. 7. Possible approaches to support Dynamic workflow requirements

human resources are used to generate the processes. In the business efficiency aspect, our approach is better because this approach requires a person to input necessary information to create the processes whenever inventories run out. Figure 7 (d) shows an ad hoc workflow approach that uses multiple work items to access other systems. In this approach, users have to manually coordinate the work items through an application at the routing point. The adaptability of this approach is quite good because the work items can be coordinated in the middle of process execution. But it is less adaptable than our framework because the coordination cannot be done automatically. Whenever a new process is started, the approach has to manually coordinate the work items. It also requires more resources, such as human resources and applications.

Table 1. Comparison of four approaches

Approaches	Automaticity	Adaptability	Business Efficiency	Physical Resources
Proposed Framework	Automatic	High	High	Medium
Conventional Workflow	Manual	Low	Low	High
E-procurement System	Manual	Medium	Medium	Medium-High
Ad hoc Workflow	Semi-automatic	Medium-High	Medium	Medium-High

Table 1 compares the approaches from several different points of view: automaticity, adaptability, business efficiency and physical resource.

6 Conclusion

In this paper, a framework supporting dynamic interoperation among workflow management systems is proposed. Multiple subprocess task (MST) is developed, and the multi-tiered dynamic state transition model is developed using objected-oriented concepts such as encapsulation, delegation, and dynamic binding. The subprocess task can represent many alternative internal or external services. By using subprocess task, the main process definition can be much simpler and the reusability of frequently-used process definition can be improved. The interoperability of two or more software components is a scalable form of reusability [14]. The interoperability of heterogeneous workflow systems might also be regarded as a scalable form of reusability and inter-organizational cooperation. It gives more chances for one workflow system to reuse the functions of other workflow system for the execution of a business process.

We showed that MST can handle a complex situation, where a task has to represent multiple SPs and the number and execution order of those SPs are not known until run-time. With MST, we also defined a framework supporting dynamic workflow interoperation consisting of four components: workflow engine, adapter, SIRs, and Wf-XML messages.

Numerous benefits can be expected from dynamic workflow interoperation. For instance, easy selection of appropriate subprocesses at runtime can improve the scalability and flexibility of workflow interoperation mechanism. We analyzed the proposed framework and we revealed that the encapsulating and dynamic binding of subprocesses of proposed framework improves automaticity, adaptability, and business efficiency with less physical resources compared to other approaches. Although we have not discussed Quality of Service (QOS), semantic description and ontology, and business transaction contract between trading partners, those topics are important features for workflow interoperability and must be handled in near feature.

References

1. Amit P. Sheth, Wil van der Aalst, Ismailcem B. Arpinar, “Process Driving the Networked Economy,” IEEE Concurrency, July-September 1999.
2. Christensen, E., Curbera, F., Meredith, G., Weerawarana, S.: Web Services Description Language (WSDL) 1.1. World Wide Web Consortium note, http://www.w3.org/TR/2001/NOTE-wsdl-20010315 (2001)
3. D. S. Han, J. Y. Shim, and C. S. Yu, “ICU/COWS: A Distributed Transactional Workflow System Supporting Multiple Workflow Types,” IEICE Transactions on Information and Systems, Vol. E83-D, No. 7, July 2000.
4. Workflow Management Coalition, "Workflow Standard - Interoperability: Abstract Specification, TC-1012", Oct. 1996.
5. J. Klingemann, “Controlled Flexibility in Workflow Management,” Proceedings of the 12th International Conference on Advanced Information Systems Engineering(CAiSE’00), Stockholm, Sweden, June 2000, pp. 126-141.
6. Ehnebuske, D., McKee, B., Rogers, D.: UDDI Version 2.04 API Specification. UDDI.org, http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.htm (2002) 5. P. Wegner, “Interactive Software Technology,” Handbook of Computer Science and Engineering, CRC Press, May 1996.
7. S. Lee, D. Han and D. Lee, “A Pattern for Managing Distributed Workflows,” 7th Pattern Languages of Programs Conference (PLoP 2000), Aug. 2000.
8. Mark Grand, “Patterns in Java volume 1: a catalog of reusable design patterns illustrated with UML,” Wiley computer publishing, 1998.
9. United Nations Economic Commission (UN/EC): Electronic Data Interchange for Administration, Commerce and Transport – Application Level Syntax Rules (ISO 9735).
10. ebXML Technical Architecture Specification v1.04, 16 February 2001; available online at (http://www.ebxml.org/specs/ebTA.pdf)
11. RosettaNet, RosettaNet Implementation Framework (RNIF) Specification, Version 1.1. 1999: available online at (http://www.rosettanet.org).
12. Workflow Management Coalition, “Workflow Standard – Interoperability Wf-XML Binding,” TC-1023, May 2000.
13. Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J-J., Nielsen, H.F.: SOAP Version 1.2 World Wide Web Consortium Recommendation, http://www.w3.org/TR/soap/.
14. P. Wegner, “Interactive Software Technology,” Handbook of Computer Science and Engineering, CRC Press, May 1996.

A Text Mining Approach to Integrating Business Process Models and Governing Documents

Jon Espen Ingvaldsen[1], Jon Atle Gulla[1], Xiaomeng Su[1], and Harald Rønneberg[2]

[1] Norwegian University of Science and Technology, Department of Computer and Information Science, Sem Saelands vei 7-9, NO-7491 Trondheim, Norway
{jonespi, jag, xiaomeng}@idi.ntnu.no
[2] Statoil ASA, 4035 Stavanger, Norway
haro@statoil.com

Abstract. As large companies are building up their enterprise architecture solutions, they need to relate business process descriptions to lengthy and formally structured documents of corporate policies and standards. However, these documents are usually not specific to particular tasks or processes, and the user is left to read through a substantial amount of irrelevant text to find the few fragments that are relevant to him. In this paper, we describe a text mining approach to establishing links between business process model elements and relevant parts of governing documents in Statoil, one of Norway's largest companies. The approach builds on standard IR techniques, gives us a ranked list of text fragments for each business process activity, and can easily be integrated with Statoil's enterprise architecture solution. With these ranked lists at hand, users can easily find the most relevant sections to read before carrying out their activities.

1 Introduction

Several organizations have made business process models and governing documents public on their intranet solutions as part of large enterprise architecture initiatives. The motivation behind such initiatives is to provide available documentation about the execution of business activities to any user in the organization, and enable users to view their task in a business process perspective.

Business process models and governing documents are, by nature, tightly coupled. Business process models show graphical representations of the order by which activities are executed. Typically, additional information like involved resources and hierarchical levels are also provided [8][16]. Governing documents, on the other hand, provide textual and full descriptions of how business processes should be executed. While business process models enable the user to grasp an overview of how activities, processes and resources are related, governing documents provide the user with operative principles and executable instructions.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 473–484, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Traditionally, elements in the graphical business process models are manually related to relevant governing documents. As goverining documents are lengthy, formally structured bodies of text and describe issues that are not directly relevant for the execution of specific business activities, it is in many cases bothersome to locate the fractions of text that are of importance.

The purpose of this paper is to describe a text mining approach that exploits the hierarchical structure of governing documents and identifies similarity relationships dynamically. Use of such approaches has several potential advantages, including the abilities to:

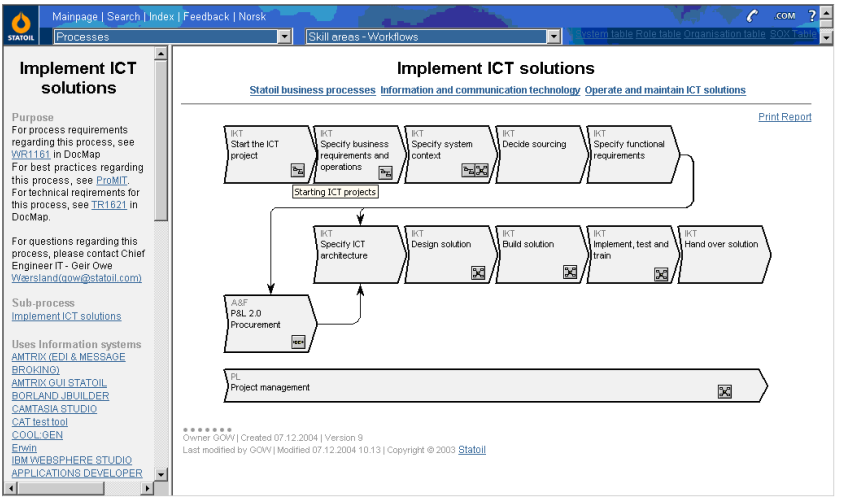
- Simplify the task of providing documentation that contains up to date and consistent hyperlinks.
- Provide ranked lists of texts relevant to a selected business process. Such lists can contain whole governing documents or be broken down to levels of sub sections or paragraphs.
- Guide users through the reading of governing documents and provide visual measures about which sections and paragraphs that are of importance.

The *Business Process Model* (BPM) is a system of accessing business process documentation at a major Norwegian oil company, Statoil ASA. We will use their system as an example and test environment for the proposed text mining approach. The BPM is breifly presented in Section 2. In section 3, we describe the text mining approach for extracting similarity relationships between model elements and fractions of governing documents. Potentials of intregrating such techniques to the BPM system is described in Section 4. Section 5 gives a presentation and discussion of results from initial attempts of applying text mining techniques to the documentation in BPM. Section 6 presents areas of related work, followed up by directions for future work and concluding remarks in Section 7.

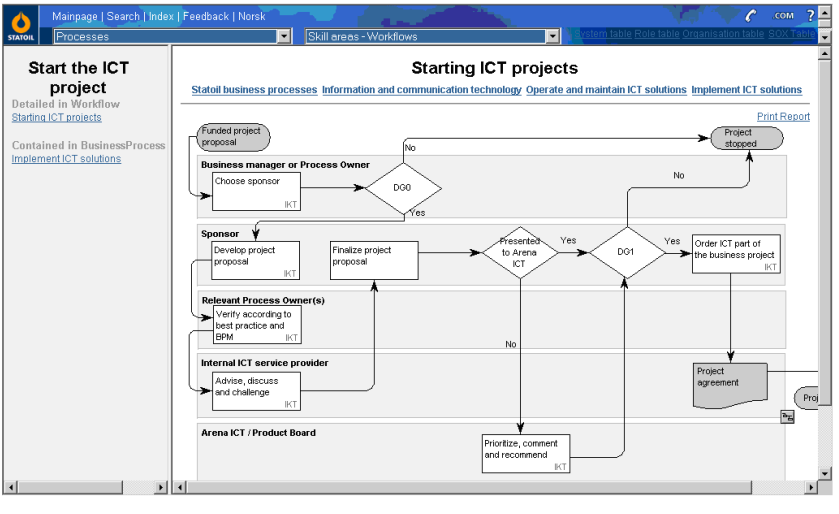
2 Business Process Models and Governing Documents in Statoil ASA

Statoil is an integrated oil and gas company with substantial international activities. It is represented in 28 countries, and the group has about 24 000 employees. Nearly 50 per cent of these employees work outside Norway. Statoil is the leading producer on the Norwegian continental shelf and is operator for 20 oil and gas fields. The company is one of the world's largest operators for offshore oil and gas activities.

BPM is used to document and communicate the relation between business processes, information and IT systems. BPM is a model, not a drawing, in which one can view relations from different angles (e.g. all IT systems related to one specific process, all processes using one specific IT system, all IT systems requiring a specific type of information). The principal rule is to connect IT systems



(a) A screenshot of the BPM system that shows a decomposed view of the "*Implement ICT solutions*" process, which is part of the main process "*Information and communication technology*". A description, inluding reference to three governing documents, for the "*Implement ICT solutions*" process is displayed on the left side of the figure.



(b) A screenshot of the BPM system that shows the decomposition of the "*Starting ICT projects*" process, in figure 1(a), into a workflow model. The uncompleted description of the process is displayed on the left side of the figure. Further descriptions of each specific activities is displayed when they are selected.

Fig. 1. Examples of screenshots from BPM

to superior business process descriptions. It is part of a comprehensive enterprise architecture initiative with which the company wants to make sure that its resources are well integrated and accounted for.

In BPM, Statoil's twelve main business processes are modelled. Each main process is decomposed into 2 - 3 layers of sub-processes. The lowest layer sub-processes are decomposed into one or more layers of workflows. A process owner is appointed for each of the twelve main business processes. This person is responsible for ensuring that BPM is correct for his own process. The business is responsible for ensuring that it acts in accordance with the process owner's business process model and workflows, including use of IT system. I.e. the BPM shall also be used to govern the relations between business processes, information and IT system ensuring the business makes use of the process owner's recommended IT systems. From an IT point of view, BPM is used to manage the portfolio of IT systems. Examples of how models and information is presented in the BPM is shown in figure 1(a) and 1(b). The model in Figure 1(b) shows how the process of implementing ICT solutions, decomposed into a sequence of workflow activities for each stage of the implementation project.

In addition to the graphical models, BPM provides additional descriptions of business processes, workflows and workflow elements like activities. A description contains typically purpose and references to IT systems, information elements and governing documents to be used by the process or workflow. As shown in figure 1(a) and 1(b), descriptions are shown on the left side of the screen.

Governing and advisory documents are part of Statoil's control system and are prepared on the basis of the group's need to control operations. They help to ensure systematic control of the group's own activities and continuous improvement work throughout the group, and as such to contribute towards greater value creation and strengthened competitiveness. These documents also describe how to deal with relevant requirements from the authorities.

As we can see in figure 1(b), BPM is not yet completed with all models and descriptions, but it is required that all business process models and workflow models shall be documented in BPM system. The model in figure 1(b) is a control flow model and specifies how the different actors in Statoil are onvolved when new ICT projects are initiated. The less completed part of the system is at the moment is the information model.

The vision of BPM is that all tools, documents, descriptions, information, etc. that are needed to carry out an activity shall be made available from BPM with a few mouse clicks.

3 Text Mining Approach

Instead of relating activities and process in the graphical business process models to relevant governing documents manually, text mining techniques can be used to autmatically identify such relationships. Text mining techniques have also abilities to establish relations to the most relevant sections and paragraphs within the governing documents and to rank these after similarity measures.

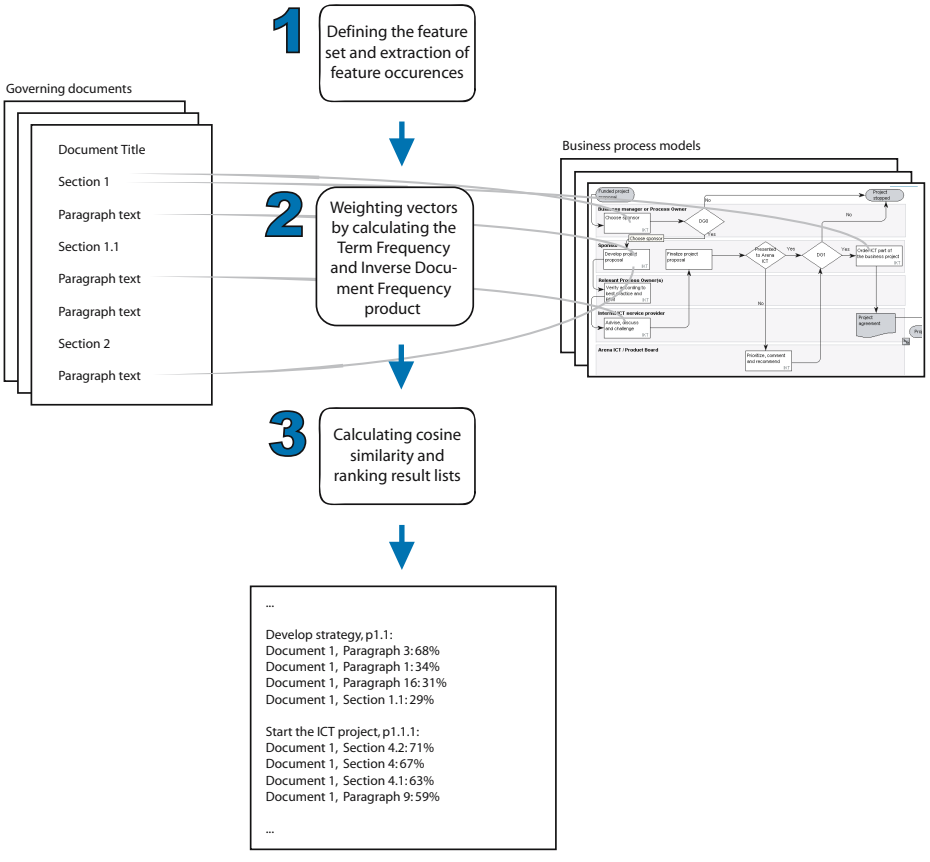


Fig. 2. Illustration of the three stages that are involved in the process of achiving an output of lists of ranked similarities

Figure 2 illustrates the stages that are involved in the proposed text mining procedure. The input to the procedure is a set of textual governing documents and a set of business process models. The output of the procedure is ranked lists that shows the fractions of text in the governing documents that are most similar to elements in the graphical business process models. In these calculations, we use the textual descriptions in the neighborhood of a model element to represent its content. We will describe each step more in detail.

The first stage consists of defining the feature set and extraction of feature occurences from textual descriptions in the business process models and governing documents. Features, in this sense, are the set of content bearing terms that will form the basis for our vector space model. We define this set of features to be the total set of base-form words that are included in any textual description of model elements in the business process models. Stop words like "*and*", "*the*", "*to*", etc. are not included.

Having the set of features specified, we are able to calculate feature vectors for sections and paragraphs in the governing documents and for each element in the business process models. Text that is given to describe elements in the business process models, typically consists of two to four words. In order to increase the potential of finding good matches in the governing documents and to get a better image of the concepts that are involved, we are also including feature occurences in neighbor, parent and successor elements. Neighbor elements are any model element, including processes, activities, events, desicion points, and documents, that have a subsequent relationship to each other. Figure 1(a) and 1(b) shows several examples of neighbor elements, like the pairwise relationships between "*Develop project proposal*" and "*Verify according to best practice and BPM*", and "*Order ICT part of the business project*" and "*Project agreement*". Parent elements are super processes that descibes the business flow on a more general level, while successor elements are components of a specific business process at a decomposed level. An example of such relationships are shown in figure 1(a) and 1(b), where every model element of figure 1(b) is a successor of the process "*Starting ICT projects*". The feature occurences in neighbor, parent and successor elements are weighted less than word occurences in the respective model element.

In order to calculate feature vectors for fractions of the governing documents, sections and paragraphs are identified and labeled with unique ids. Documents, sections and paragraphs form a hierarchical structure, where documents consist of a title and a set of sections and sections consist of a title, subsections and paragraphs. The paragraphs are the leaf elements in this structure and consist of a unbroken sequence of text. An typical example of this hierarchical structure is shown in figre 3 where Section 4 consists of a title, *"Report to IT project Inventory"*, and four sub sections. Further, each sub section consist of a title and one or two paragraphs. Feature vectors for documents, sections and paragraphs are calculated by counting the number of feature occurences within the text that they contain.

The second stage in the procedure consist of applying a weighting schema that enable a fair discrimination of involved elements. For all of the feature vector calculations, the Inverse Document Frequency (*idf*) is applied. *idf* can be interpreted as the informative value of the feature and is defined as follows [1]:

Definition 1. *Let N be the total number of documents in the systems and n_i be the number of documents in which the index term k_i appears. The inverse document frequency for k_i, idf_i is given by*

$$idf_i = \log \frac{N}{n_i} \tag{1}$$

Then, the total term weighting scheme is given by

$$w_{i,j} = (f_{i,j,prim} + \alpha \times f_{i,j,parent} + \beta \times f_{i,j,successor} + \gamma \times f_{i,j,neighbor}) \times \log \frac{N}{n_i}, \tag{2}$$

where $f_{i,j,prim}$ is the frequency of terms appearing in the main model element or governing document text, $f_{i,j,parent}$ is the frequency of terms appearing in parent

model elements, $f_{i,j,successor}$ is the frequency of terms appearing in successor model elements, and $f_{i,j,neighbor}$ is the frequency of terms appearing in neighbor model elements. α is the weight that is applied for parent model elements, β is the weight that is applied for successor model elements, and γ is the weight that is applied for neighbor model elements.

In the last stage, the similarity between feature vectors that represent model elements and feature vectors that represent fractions of the governing documents are estimated and ranked. Similarity in vector space models is determined by using associative coefficients based on the inner product of two feature vectors, where feature overlap indicates similarity. This inner product is usually normalized. The most popular similarity measure is the cosine coefficient, which measures the angle between two vectors. That is

$$\cos\theta = \frac{\mathbf{v} \bullet \mathbf{w}}{\|\mathbf{v}\| \, \|\mathbf{w}\|}, \tag{3}$$

where $\mathbf{v} \bullet \mathbf{w}$ is the inner product of the two vectors, and $\|\mathbf{v}\|$ is the Euclidean length of a vector $\mathbf{v}$.

The percentual similarity measures in the ranked lists of figure 2 are found by calculating the angle θ, inverse relative to the maximum angle of 90°.

4 Accessing Model-Related Textual Information

Many governing documents describe issues related to the execution and specific issues of business processes on a general level. As a consequence, the execution procedure for sub processes or activities are not described precisely in individual sections. However, this does not mean that specific sections and paragraphs have the same relevance for involved model elements. By having the set of similarity measures available we are able to pinpoint those fractions of text that have most relevance for the execution of a specific business process or activity.

In order to fulfil the potential advantages of providing both ranged lists of relevant texts and guidance through the reading of governing documents, several issues related to presentation of information are of concern.

The most simple approach for presenting ranked list of hyperlinks to locations in governing documents is to include any fraction of text that either are within the top-n most similar items or have a similarity measure greater that a certan static threshold level. An example of such a ranked list is shown in table 1. Conklin [2] points out that hyperlinks are found to be more useful if users are able to dynamically filter out those that satisfies certain properties. In our case, we see a need for such filtering functionality where the filtering properties include fraction level (*document*, *section* or *paragraph*), governing document category, and similarity threshold.

There are several alternatives for presenting similarity measures in the governing documents. They can be shown as percentual numbers in the margins of the texts, icons representing specified similarity categories (high, medium, low),

Table 1. Examples of a resulting ranked list of text for the process *"Start the ICT project"*

Relevant text for *"Start the ICT Project"*		
No.	Item	SM
1	Document #1, Section 4.2	71 %
2	Document #1, Section 4	67 %
3	Document #1, Section 4.1	63 %
4	Document #1, Paragraph 9	59 %
5	Document #1, Section 4.3	58 %
5	Document #1, Paragraph 31	36 %

4 Report to IT Project Inventory

4.1 Project Inventory
§28 Project Inventory is the ICT Arena's overview of all ICT-projects in Statoil. All projects with an ICT solution exceeding USD 60000 are to report to Project Inventory throughout the project phases. It is the responsibility of the Project Manager to ensure this reporting, but still the Sponsor is accountable for the reporting.

4.2 Report project proposals
§29 As part of the process described in the "Start ICT project" chapter, all new project proposals are to be reported to the Arena ICT. This is normally done as part of the annual budget process. The project proposal information reported to Project Inventory is the basis for the prioritization recommended by the Arena ICT prior to DG1 (Project Sponsor is accountable for the reporting).

4.3 Report project and status information
§30 After DG1, project and status information are to be reported to Project Inventory on a monthly basis up to DG5. As part of DG5, the Project Manager reports to Project Inventory whether sponsor accepts delivery.

4.4 Report compliance with Statoil architecture
§31 The purpose of this reporting is to report compliance with the ICT architecture of Statoil. This reporting is in addition to the delivery parameters specified by sponsor. It is the responsibility of the Project Manager to ensure this reporting, but still the Sponsor is accountable for the reporting.

§32 The reporting includes the following KPI's:
- Is the technical architecture in line with all governing documents for ICT?
- Is the solution architecture in line with all governing documents for ICT?
- Is the information architecture in line with all governing documents for ICT?
- Is the business architecture verified with all relevant process owners and BPM is updated?

Fig. 3. An example of how the text items in table 1 can be visualized for a specific document. Sections and paragraphs are given a background color that is based on their relevance for the business process "*Start the ICT project*". As we can see, Section 4.2 is found to be most relevant. The number that is given initially at each paragraph is the unique paragraph identifier for the respective paragraph.

by use of background color, etc. An example of how similarity measures can be displayed as background color depths is shown in figure 3. In this case the color depth is decided based on the similarity value and document structure. Subsections or paragraphs are shown with a darker background color if their similarity measure is larger than the similarity measure of their super section.

5 Discussion and Results

To test out the applicability of the presented text mining approach in a real-world environment, business process models describing the main process "Information and communication technology" at four decomposed levels and related governing documents were applied.

Content bearing terms were counted for every element in the business process models and each section and paragraph in the governing documents. These basis values were fed into an implemented Java application that created weighted feature vectors. As feature vectors for the model elements were calculated, terms that appeared in parent and successor elements were weighted by $\alpha, \beta = 0.8$, while appearences in neighbor elements were given a weight $\delta = 0.5$.

The resulting lists of ranked text fractions included both very good and poor matches, but the overall impression was promising. Those model elements that found the best matches had, typically, all of its content bearing terms present in the related texts. The relationship from the process *"Start the ICT project"* to Section 4.2 in Document #1 had a similarity measure of 71%. As we see in figure 3, the main reason for this high scoring value is the presence of the highly weighted terms *"Start"*, *"ICT*, and *"project"*. This example shows also the presence of noise, as these terms are present in a sentence that refers to another section.

Models elements that resulted in lists with low similarity measures had, typically, none of its content bearing terms present in any sections or paragraphs. For all instances where this situation occured, the relationships to fractions of text were given based on the description of neighborhood elements.

The presented approach can be extended on several areas in order to improve the results, including:

- Make use of domain ontologies to capture the similarity between concepts.
- More extensively exploit the hierarchical structure of the governing documents when we are weighting the importance of content bearing terms, i.e., give terms that are present in the title of documents or sections a higher weight than terms that are present in the paragraphs.
- Involving tagging for disambiguation, thus making the system able to differentiate between verbs and nouns that are equally spelled.

6 Related Work

Works related to the issues discussed in this paper are in the area of relating ontologies and documents. The relationship between ontologies and documents are

two folded. On the one hand documents are used to aid the construction of domain ontologies. On the other hand, ontologies are used to annotate documents. The former is called ontology learning and the latter ontology annotation. We will discuss them in turn.

The usefulness of domain ontologies has been widely acknowledged, especially in relation to the Semantic Web [11][5]. A critical issue here is the automatic construction of ontologies. To ease the task of ontology construction, machine learning and automated language processing techniques are often used to extract concepts and relations from structured and unstructured text. The OntoLearn system [15], for instance, extracts relevant domain terms from a corpus of text, relates them to appropriate concepts in WordNet [13], and detects taxonomic and other semantic relations among the concepts. Similar works in this line are reported in [10][14][4].

Ontology annotation, on the other hand, deals with using existing ontologies to annotate documents in order to provide richly interlinked, machine-understandable data. The KA2 initiative [9] aimed at providing semantic retrieval of a knowledge portal. The potential users provide knowledge in a decentralized manner, e.g. by annotating their web pages in accordance to the common ontology. The knowledge is then collected at the knowledge portal by crawling and presented in a variety of ways. In relation to the Semantic Web, a number of projects involve knowledge markup in the Semantic Web, viz. SHOE [7], OntoBroker [3], CREAM [6], WebKB [12].

In our work, the domain process model is available through some other modeling efforts. The governing documents are associated with relevant model elements as part of the organizational knowledge assets. The original contribution of the work presented in this paper is related to the idea of taking advantage of the availability of the model and the availability of the associations between model elements and governing documents to help the users gain a fast and precise understanding of the work process. It is achieved by indicating relevance of textual descriptions to the process model elements at a finegrained level, i.e., at section and paragraph levels instead of the whole document level.

The issue of using relevant documents to build feature vectors has been studied in [17]. In this work, the governing documents that are associated to a model element are used as building material to construct feature vectors of the model element. Text mining techniques are employed for the construction of such feature vectors for both the model elements and the paragraphs. In such a way, the business process model and the governing documents are better integrated so that the user can relate model elements with paragraphs of governing documents and vice versa.

7 Conclusion and Future Work

With the BPM solution, Statoil intends to improve the coordination and utilization of its internal resources. The idea is to describe in detail how business processes are to be carried out, which software and hardware is needed, and

which internal people or departments are involved. The company's governing documents describe the rules and standards for all internal activities, and each process has to be executed in accordance with the decisions recorded in the governing documents.

Due to the length and formal structure of governing documents, though, it is not easy for users to consult the governing documents before carrying out a particular task. Most of the text is not relevant, but they user cannot know where to look before having read the whole document.

Use of text mining techniques to integrate governing document and graphical business process models enable dynamic and precise linking between the two medias. Users can pick out the text elements most relevant to the task at hand. The user does not need to read the whole document, but can consentrate on the paragraphs and sections that pertain to the task he is working on. The current approach makes use of lemmatized textual descriptions and a weighting scheme for word occurences in related model elements.

The approach presented in this paper is one of several information retrieval techniques that can be applied to facilitate the content of business process documentation. Future work includes refinements of the text mining techniques of this paper. Such refinements includes more sophisticated weighting schemas and involvement of linquistical analyses and ontology enrichments.

References

1. R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.
2. J. Conklin. A survey of hypertext. Technical Report 2, Austin, Texas, 3 1987.
3. S. Decker, M. Erdmann, D. Fensel, and R. Studer. *Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information*. Kluwer Academic Publisher, Boston, 1999.
4. R. Engels and T. Lech. *Towards the Semantic Web: Ontology-Driven Knowledge Management*, chapter Generating Ontologies for the Semantic Web: OntoBuilder. John Wiley & Sons, 2003.
5. D. Fensel, F. van Harmelen, I. Horrocks, D. L. McGuinness, and P. F. Patel-Schneider. Oil: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems*, 16(2):38–45, 2001.
6. S. Handschuh, S. Staab, and A. Maedche. Creating relational metadata (cream) - a framework for semantic annotation. In *The Emerging Semantic Web*, 2001.
7. J. Heflin and J. Hendler. Searching the web with shoe, 2000.
8. J. E. Ingvaldsen, J. A. Gulla, O. A. Hegle, and A. Prange. Revealing the real business flows from enterprise systems transactions. *Accepted for publication, 7th International Conference on Enterprise Information Systems*, 2005.
9. A. Maedche, H. p. Schnurr, M. Erdmann, and S. Staab. From manual to semi-automatic semantic annotation: About ontology-based text annotation tools, June 23 2000.
10. A. Maedche and S. Staab. Mining ontologies from text. *Lecture Notes in Computer Science*, 1937:189–202, 2000.
11. E. Maedche and S. Staab. Ontology learning for the semantic web, Feb. 08 2001.

12. P. Martin and P. Eklund. Embedding knowledge in Web documents: CGs versus XML-based metadata languages. *Lecture Notes in Computer Science*, 1640:230–??, 1999.
13. G. Miller. WordNet: an online lexical database. *International Journal of Lexicography*, 3(4), 1990.
14. M. Missikoff, R. Navigli, and V. Velardi. The usable ontology: An environment for building and assessing a domain ontology. *Lecture Notes in Computer Science*, 2342:39–53, 2002.
15. R. Navigli, P. Velardi, and A. Gangemi. Ontology learning and its application to automated terminology translation. *IEEE Intelligent Systems*, 18(1):22–31, 2003.
16. A.-W. Scheer and M. Nüttgens. ARIS architecture and reference models for business process management. *Lecture Notes in Computer Science*, 1806:376–389, 2000.
17. X. Su and J. A. Gulla. Semantic enrichment for ontology mapping. In *Proceedings of the NLDB*, pages 217–228, 2004.

A Process-Driven Inter-organizational Choreography Modeling System

Kwang-Hoon Kim

Collaboration Technology Research Lab.,
Department of Computer Science, Kyonggi University,
San 94-6 Yiuidong, Youngtongku Suwonsi Kyonggido,
South Korea, 442-760
kwang@kyonggi.ac.kr
http://ctrl.kyonggi.ac.kr

Abstract. Currently we have been developing a process-driven e-business service integration (BSI) system through functional extensions of the ebXML technology. And it is targeting on the process-driven e-business service integration markets, such as e-Logistics, e-SCM, e-Procurement, and e-Government, that require process-driven multi-party collaborations of a set of independent organizations. The system consists of three major components - *Choreography Modeler*, *Runtime & Monitoring Client* and *EJB-based BSI Engine*. This paper particularly focuses on the choreography modeler that provides the modeling functionality for ebXML-based choreography and orchestration among engaged organizations in a process-driven multiparty collaboration. Now, it is fully operable on an EJB-based framework environment (J2EE, JBOSS, and Weblogic), and also it is applied to e-Logistics process automation and B2B choreography models of a postal service company. This paper mainly describes the implementation details of the modeler, especially focusing on modeling features of the process-driven multi-party collaboration functionality.

Keywords: Inter-organizational e-business service integration, B2B choreography and orchestration, Dual-party collaboration model, Process-driven multi-party collaboration model, Inter-organizational choreography modeling system.

1 Introduction

Recently, electronic commerce and its related technologies have been swiftly adopted and hot-issued in the real world. This atmosphere booming e-commerce is becoming a catalyst for triggering explosion of the electronic logistics and supply chain management technologies and markets as well. This also means that organizations are much closer each other and need to support the inter-organizational cooperative activities with great efficiencies. We need to give a special attention on ebXML in order to implement the process driven e-business service integration. Recently, ebXML is a crucial technology and standard for

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 485–494, 2005.
© Springer-Verlag Berlin Heidelberg 2005

e-commerce. The vision of ebXML is to create a single global electronic marketplace where enterprizes, without any concerns about size and geographical location, can meet and conduct their businesses with each other through the exchanges of XML-based messages. Therefore, ebXML technology enables organizations to do business with anyone in anyplace over the internet.

It, however, does not seamlessly provide the complete solutions for the process-driven trading procedures of a serial of associated organizations. It means that we need to fortify ebXML against the process-driven choreography and orchestration between multiple party collaborations that are able to bring organizations much closer and make them much more tightly coupled by supporting inter-organizational activities. But, the process driven e-Business service integration and choreography models are a little different from the traditional B2B model of e-Commerce in terms of the behavior of choreography among organizations in collaboration. That is, in the traditional B2B e-Commerce model, only dual-party (buyer/seller) organizations collaborate along with the direction of CPA/CPP of ebXML, which is established through agreement between the participating parties. It, however, does not seamlessly provide the complete solutions for the trading procedures. This means that we need to fortify ebXML against the process-driven choreography and orchestration of multiple party collaborative organizations. Without some further modification, the ebXML technology won't be directly fit into those process-driven e-Business service integration domains. So, in this paper, we try to extend the ebXML technology's modeling functionality through proposing a process-driven multi-party collaboration model and its modeling components. This paper describes the implementation details of the choreography modeling system and presents its operational examples.

In the next sections, it presents the backgrounds and related works that have been done in the literature, and simply introduces the overall structure of the process-driven e-Business service integration system and its application example - e-Logistics of a cyber-shopping mall. The main section of this paper consists of descriptions of process-driven choreography model, system components and their relationships, class-diagram and usecase diagram of the modeling tool. Finally, we introduce a serial of the modeler's screen captures to show the practical applications of the modeler.

2 Related Works

In South Korea, workflow/BPM and its related technological fields, such as B2B e-Commerce, ERP, SCM, CALM, EAI, etc., begin attracting great attention from the society of information science and database management in aspects of not only research topics but also industrial one such as information application fields. They really do catch public attentions. There are several projects ongoing research and development of workflow and BPM systems issued by universities and research institutes, and even by industry as well. Of course, these technologies are issued and fairly settled down in the worldwide information technology arena, too. We, in this paper, are particularly interested in the B2B e-Commerce

process modeling technologies. As a matter of fact, in order to accomplish the total process automation for B2B e-Commerce, it is necessary for the workflow and BPM technology to be integrated, as a platform, with major four contributory technologies: object-orientation, EAI, web service, and XML technology. So, there might be two possible approaches, as we can imagine, to deploy a total process automation entity for B2B e-Commerce - open e-business framework (ebXML) and closed e-business framework (inter-organizational workflow and BPM [13]) approaches. The open e-business means that the business contractors or partners of the B2B e-business can become anyone who are registered in the registry/repository, in contrast to that they are predetermined in the closed framework approach. We concluded that the former is more reasonable approach for the process-driven e-Business service integration domain.

We would characterize ebXML as an open e-business framework because it is targeting on opening a business contract between universal organizations unknown each other. The ebXML is a set of specifications that enable to form a complete electronic business framework. Suppose that the Internet is the information highway for e-businesses, then the ebXML can be thought of as providing the on-ramps, off-ramps, and the rules of the road,[3] and it is proposed by a joint initiative of the United Nations (UN/CEFACT) and OASIS, which are developed with global participation for global use. This partnership brings a great deal of credibility to ebXML being representative of major vendors and users in the IT industry and being supported from leading vertical and horizontal industry groups including RosettaNet, OTA (Open Travel Alliance) and many others. Membership in ebXML is open to anyone, and the initiative enjoys broad industry support with hundreds of member companies, and more than 1,400 participants drawn from over 30 countries.[11]

We would not describe the details of ebXML [3] in here, but adopt the basic concept and mechanism of ebXML as an e-business service agent providing process-related information to the e-business service entities over organizations. But, the ebXML is hardly applicable to e-Logistics management or supply chain management framework, without some modifications, because it is basically proposed for e-business contracts between two parties (buyer and seller), each of which corresponds to an organization in collaboration. That is, the concept of business process being accepted in the ebXML domain is completely different from the concept in the e-Logistics domain. So, we try to extend its modeling functionality so as to be reasonably adopted in those process driven e-business service integration systems.

3 The Process-Driven Choreography Modeling System

In this section, we at first define the basic concept of process driven e-Business service integration (BSI) and choreography model that is a little different from the traditional dual-party B2B model of e-Commerce in terms of the behaviors of collaborations among organizations. And we describe the functional details of the choreography modeling system, such as its overall system architecture and

its major components, which is extended from the ebXML basic functionality so as to cope with the process-driven choreography model.

3.1 The Process-Driven Inter-organizational Choreography Model

The process-driven e-business service integration and Choreography model is the specifications of business contracts and of how to integrate the business services between organizations that are engaged in a process-driven multiparty collaboration. Fig. 1 shows a simple example of a dual-party collaboration model consisting of three business transactions, and each business transaction performs two business activities - *Requesting activity* and *Responding activity* - that are conducted by the initiating role's organization and the responding role's, respectively. This dual-party collaboration model becomes a binary collaboration model that represents a member of performs on a process-driven e-business service integration and choreography model as shown in Fig. 2.

Fig. 2 is to graphically represent a process-driven e-business service integration and choreography model. The model consists of three dual-party collaborations (Binary Collaboration 1 $\sim$ 3) that have a certain type of control-precedence relationships, each other, such as sequential, disjunctive or conjunctive relationship, according to the business contracts of the organizations in multiparty collaboration. And the dual-party collaboration is specified by the activity flow diagram, in which each activity represents a business transaction. Also, each organization (Organization A $\sim$ D) has associated to either the initiating role or the responding role, or both of the roles like Organization B.

Additionally, each dual-party collaboration in the model is represented by the ebXML specifications. The ebXML's information models define reusable components that can be applied in a standard way within a business context, and enable users to define data that are meaningful to their business and also maintaining interoperability with other business applications. Also, the ebXML messaging service specification defines a set of services and protocols that enables electronic business applications to exchange data. The specification allows any level of application protocols including common protocols such as SMTP, HTTP, and FTP to be used. The Collaborative Partner Agreement defines the technical

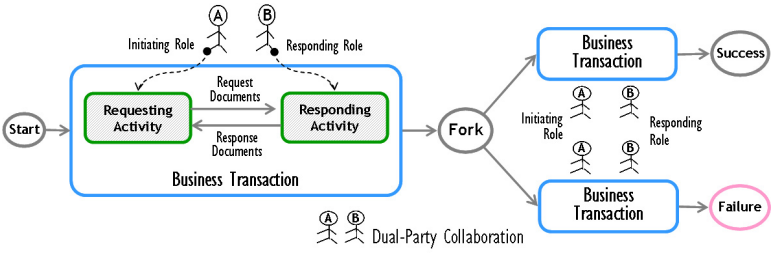


Fig. 1. A Dual-party Collaboration

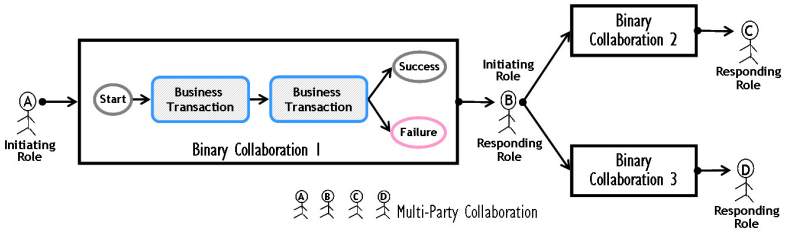


Fig. 2. A Process-driven (Multi-party) Collaboration

parameters of the Collaborative Partner Profiles (CPP) and Collaborative Partner Agreements (CPA). This captures critical information for communications between applications and business processes and also records specific technical parameters for conducting electronic business.

Finally, the process-driven e-Business service integration and choreography model is fundamentally based upon the open e-business framework as we shortly mentioned in the previous section. So, we need the Registry and Repository for the user applications in order to store company profiles and trading partner specifications. These mechanisms give access to specific business processes and information models to allow updates and additions over time. For the application developer it will store not only the final business process definitions, but also a library of core components. Especially, the ebXML Registry provides a set of services that enable sharing of information between interested parties for the purpose of enabling business process integration between such parties based on the ebXML specifications. Such information is used to facilitate ebXML-based Business-to-Business (B2B) partnerships and transactions. As a result, a set of registry services for the process-driven e-Business service integration models which provides accessibility of registry contents to clients of the Registry is defined through the ebXML Registry Services Specification.

3.2 Design of the Process-Driven Choreography Modeler

The choreography modeler is a registry-based business service integration and choreography modeling tool to be tightly coupled with the registration and register client for process-driven e-business service integration models. So, the modeler can be characterized by the concept of registry that is exactly same to ebXML's. And it is implemented by Java language and EJB framework approach so as to be deployed on various platforms without any further technical consideration. Now, the system is fully deployable and operable on EJB-based computing environment. Fig. 3 shows the functional components of the choreography modeler. Especially, the components inside of the dot-lined box in the figure are the core part of the modeler. So, the implementation details of them are described in this section.

Before describing the implementation details of the core components, we would explain the design details of the modeler. In order to implement the modeler, it is necessary to use a software development methodology. We adopt the

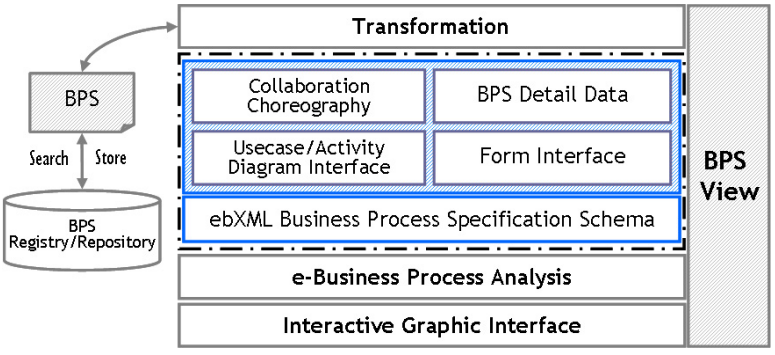


Fig. 3. Functional Components of the Choreography Modeler

MARMI-3 software development methodology that is a typical object-oriented design methodology using the usecase-based incremental approach of the spiral software development process model. In this methodology, it is very important to generate a set of usecases through analyzing the requirements of the modeler. Fig. 4 is the usecase diagram of the modeler. Based upon the usecase diagram, we are able to generate the following 19 usecases for the modeler: BP Modeling usecase, BP Modeling by UML Interface, BP Modeling by Form Interface, Business Collaboration Modeling, Role Modeling, Document Modeling, Business Transaction Modeling, Business Transaction Activity Modeling, Choreography Modeling, Start Modeling, Transition Modeling, Fork Modeling, Join Modeling, End Modeling, BPS Parsing, BP XML Viewing, BPS Storing and BPS Searching/Retrieving usecase.

Through analysis of the usecase diagram of the modeler, we are able to define a set of classes and their relationships. Fig. 5 presents the package diagram of the choreography modeler. The modeler's package diagram is composed

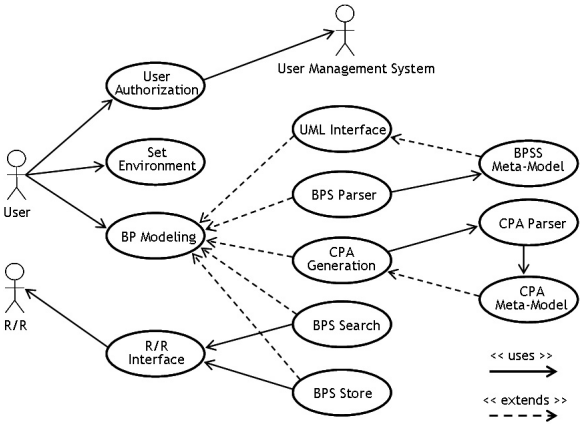


Fig. 4. The Use Case Diagram of the Choreography Modeler

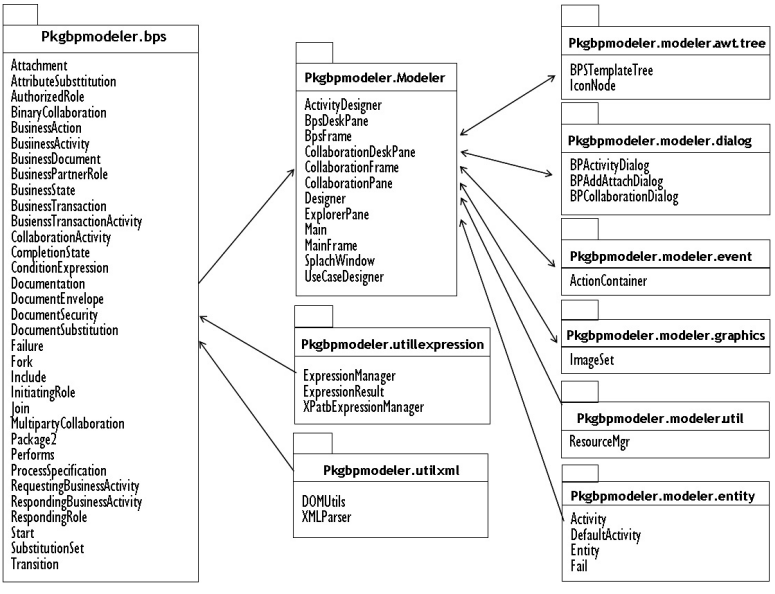


Fig. 5. The Package Diagram of the Choreography Modeler

mainly of four components - pkgbpmodeler.bps, pkgbpmodeler.model, pkgbpmodeler.utilexpression, and pkgbpmodeler.utilxml. The component, pkgbpmodeler.bps, is for the specification (or attributes) of a business process that should be provided for properly representing a process-driven e-business process integration model, and it uses the component, pkgbpmodeler.utilexpression, to define the expressions needed in the model. Also, the attributes in the specification will be finally transformed into the XML-based data format through the component, pkgbpmodeler.utilxml. The component, pkgbpmodeler.model, represents the graphical user interface of the modeler in order to provide a convenient way of defining a process-driven e-business process integration model. It also uses several utility components, such as pkgbpmodeler.model.awt.tree, pkgbpmodeler.model.dialog, pkgbpmodeler.model.event, pkgbpmodeler.model.graphics, pkgbpmodeler.model.util, and pkgbpmodeler.model.entity.

3.3 Implementation of the Process-Driven Choreography Modeler

Based upon the design outputs such as the usecase diagram, the class diagram, the business process specification schema and so on, we implemented the modeler. Through the modeler's graphical user interfaces, users are able to build their own models supporting not only the dual-party collaboration and choreography but also the process-driven multi-party collaboration and choreography. Fig. 6 depicts the modeler's graphical components and their meanings. Users are able to easily perform the modeling works by simply clicking and dragging the icons.

Icons	Meanings	Icons	Meanings
	Authorized Role/ Business Partner Role	A	Business Activity
B T A	Business Transaction Activity	F	Fork
Performs	Performs	J	Join
S	Start		Transition
			Success
			Failure

Fig. 6. Graphical Notations of the Choreography Modeler

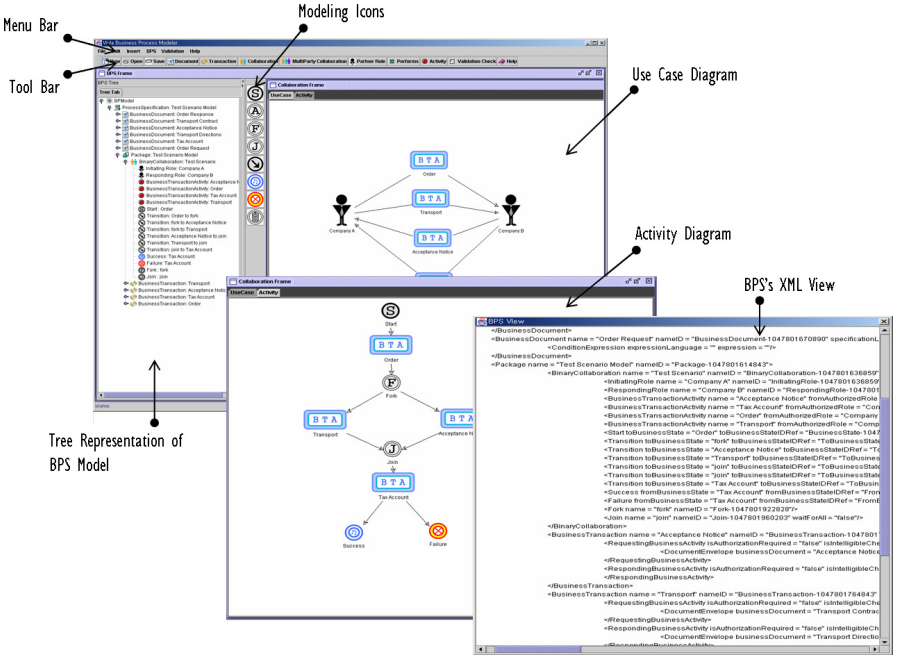


Fig. 7. An Operational Example of Dual-party Collaboration and Choreography Modeling

Fig. 7 and Fig. 8 are captured screens of the modeling examples of the dual-party collaboration and choreography model between two collaborative organizations and the process-driven multi-party collaboration and choreography model among several organizations, respectively. The window on the right-hand side windows of the screens represent XML-based specifications of both of the models. The left-hand side windows of the screens are showing a use-case diagram and an activity diagram of both a dual-party and a process-driven multi-party e-business service integration and choreography models. As you can see on the

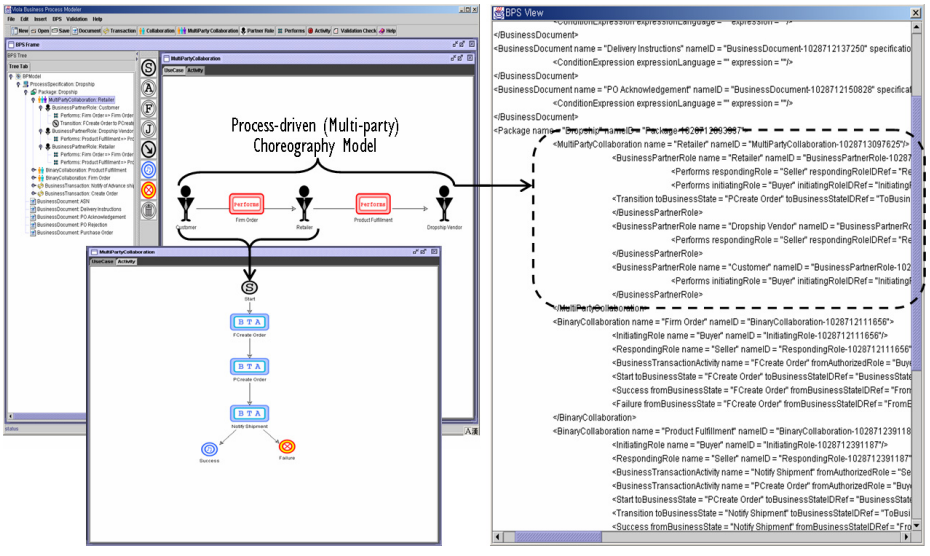


Fig. 8. An Operational Example of Process-driven Multi-party Collaboration and Choreography Modeling

top of the screens, there are several buttons, and one of them is to define a multi-party (or process-driven) e-business service integration and choreography model.

4 Conclusions

So far, we have described the implementation detials of the process-driven inter-organizational choreography modeling system. And it is targeting on the process-driven e-business service integration markets, such as e-Logistics, e-SCM, e-Procurement, and e-Government, that require process-driven multi-party collaborations of a set of independent organizations. The process-driven choreography models specifying the dual-party collaboration and the process-driven multi-party collaboration between organizations are transformed into the ebXML-based specifications, and finally the corresponding e-business service integration (BSI) system modeling system is able to enact and control the choreography models.

Not only in South Korea but also in world-wide research arena, process-driven e-business services and their related technological fields, such as e-SCM, e-Procurement, e-Logistics, e-Government, are catching great attentions from the industry of information management and business process management fields. So, there are a lot of challenges to develop and commercialize e-business solutions. This paper should be one of those active attempts for pioneering process-driven inter-organizational choreography systems toward supporting cross organizational e-business processes and service integrations. As our further research

works, we are going to implement the complete set of the cross-organizational e-business processes and service integrations system's components.

References

1. UN/CEFACT and OASIS - ebXML Registry Information Model v1.0, May (2001)
2. UN/CEFACT and OASIS - ebXML Registry Services Specification v1.0, May (2001)
3. UN/CEFACT and OASIS - Enabling Electronic Business with ebXML, December (2000)
4. Madhu Siddalingaiah: Overview of ebXML, Sun Microsystems, August (2001)
5. Haruo Hayami, Masashi Katsumata, Ken-ichi Okada: Interworkflow: A Challenge for Business-to-Business Electronic Commerce, Workflow Handbook 2001, WfMC, October (2000)
6. The JAVATM 2 Enterprise Edition Developer's Guide, Version 1.2.1, Sun Microsystems, May (2000)
7. Enterprise JavaBeans Programming, Revision A.1, Sun Microsystems, May (2000)
8. Fosdick, H.: The Sociology of Technology Adaptation, Enterprise System Journal, September (1992)
9. Kwang-Hoon Kim, Clarence A. Ellis: A Framework for Workflow Architectures, University of Colorado, Department of Computer Science, Technical Reports, CU-CS-847-97, December (1997)
10. Dong-Keun Oh, Kwang-Hoon Kim: An EJB-Based Database Agent for Workflow Definition, Journal of Korean Society for Internet Information, Vol.4, No.5, December (2001)
11. Kwang-Hoon Kim, Clarence A. Ellis: Performance Analytic Models and Analyses for Workflow Architectures, Information Systems Frontiers, Vol. 3, No. 3, (2001) 339–355
12. Sung-Su Sim, Kwang-Hoon Kim: Workcase based Very Large Scale Workflow System Architecture, Porceedings of Korea Database Society, October (2002)
13. Haruo Hayami, Masashi Katsumata, Ken-ichi Okada: Interworkflow: A Challenge for Business-to-Business Electronic Commerce, Workflow Handbook 2001, WfMC, October (2000)
14. Dogac, A., Tambag, Y., Pembecioglu, P., Pektas, S., Laleci, G. B., Kurt, G., Toprak, S., Kabak, Y.: An ebXML Infrastructure Implementation through UDDI Registries and RosettaNet PIPs, ACM SIGMOD International Conference on Management of Data, Madison, Wisconsin, USA, June (2002)
15. Jae-Gak Hwang, Se-Won Oh, ,Yong-Jun Lee: e-Logistics Integration System based on ebXML specifications, Proceeding of Korea Information Science Society, Vol 29, No 2, October (2002)
16. Dong-Keun Oh, Jong-Mi Chun, Jung-Sun Hong, Se-Won Oh, Jae-Gak Hwang, Yong-Jun Lee, Kwang-Hoon Kim: ebXML-Based e-Logistics Workflow Engine Architecture, Proceedings of Korean Society for Internet Information, Vol 3, No.3, November (2002) 348–351

A Petri Net Based Approach for Process Model Driven Deduction of BPEL Code

Agnes Koschmider and Marco Mevius

Institute of Applied Informatics and Formal Description Methods,
Universität Karlsruhe (TH),
76187 Karlsruhe, Germany
{koschmider, mevius}@aifb.uni-karlsruhe.de

Abstract. The management of collaborative business processes refers to the design, analysis, and execution of interrelated production, logistics and information processes, which are usually performed by different independent enterprises in order to produce and to deliver a specified range of goods or services. The effort to interconnect independently developed business process models and to map them to process-implementing software components is particularly high. The implementation of such collaborative inter-organizational business process models is assisted by so-called choreography languages that can be executed by software applications. In this paper, we present a Petri net based approach for process-model driven deduction of BPEL code. Our approach is based on a specific type of high-level Petri nets, so-called XML nets. We use XML nets both for modeling and coordinating business processes implemented as Web services and for deriving BPEL elements of the Web service based components. Our approach provides a seamless concept for modeling, analysis and execution of business processes.

1 Introduction

Business processes have to be designed and coordinated in a way such that a given set of objectives will be achieved. The major task of business process modeling is the representation of alternative process designs by problem-specific formal or semiformal process models. An assessment of the alternative process designs in terms of total process cost and service levels is actively to be supported. On the basis of cost and service level information, gained through process analysis, one of the design proposals is selected and finally implemented. Web services offer a suitable framework for reusing legacy systems and composing loosely-coupled services to a new organizational or inter-organizational business application.

For business process modeling different languages have been proposed, most of which are based on textual programming languages or graphical notations such as Petri nets, state charts, EPCs, BPMN or related notations [7, 9, 18]. Novel orchestration and choreography languages such as BPEL [5], BPML [2] or ebXML focus on tracking and executing collaborative business processes by business applications. All those methods and applications do not provide integrated concepts to model, analyze and execute inter-organizational collaborative business processes on the basis of Web services. Moreover a seamless concept for process modeling,

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 495 – 505, 2005.
© Springer-Verlag Berlin Heidelberg 2005

analyzing and execution is still missing. Interleaving different Web services to create business processes is the goal of BPEL4WS (short: BPEL) in order to execute business processes. BPEL enables users to compose and orchestrate services to perform certain tasks, but do not yet support analysis methods for business processes. [8] propose a Petri net semantics for BPEL to enable verification of BPEL.

Motivated by the increasing importance of the XML standard for inter-organizational document exchange, description and manipulation, languages for XML documents were developed and integrated into the Petri net formalism by so-called XML nets [13].

In this paper we present a novel approach for choreography of Web services by so-called Web service nets which are based on a special type of high-level Petri nets called XML nets. We apply Petri nets both for modeling and coordinating processes implemented as Web services and for deriving BPEL elements of the Web service based components. Functionalities that will be provided by Web services may be identified as a subnet of an XML net or can be described by the overall business process. XML nets do not only provide significant advantages with respect to modeling inter-organizational business processes; their formal foundation allows straightforward simulation and further formal analysis. Moreover, with XML nets modeled and analyzed BPEL code can be executed by Web services.

The paper is structured as follows: in the next section we present an introduction to modeling inter-organizational collaborative business processes with Petri nets. In Section 3, we will define Petri net based Web service description. In Section 4, we will present specific concepts of Petri nets and their corresponding representation as BPEL elements. The paper concludes with a summary of advantages of the presented approach and a brief outlook on future research.

2 Modeling Inter-organizational Collaborative Business Processes with Petri Nets

Petri nets [15] constitute a formal graphical process description language that combines the advantages of graphical representation with a formal semantics of behavior. Petri nets consist of static components (places, depicted by circles) and dynamic components (transitions, depicted by rectangles) that are connected by arcs. A wide range of Petri net types has been proposed in the last four decades, distinguished by different abstraction levels concerning the marking of places. The marking assigns to each place a (possibly empty) set of objects whose representation ranges from undistinguishable tokens to complex structured documents. For illustration we present a Place/Transition net (P/T net) [16] that describes the following three-layered inter-organizational collaborative manufacturing process (Fig. 1): The assembler (third layer) instantly processes incoming “end-customer orders”. The “end-customer orders” are passed to at least three manufacturers. All manufacturers should respond within a pre-defined time interval. Because of business confidentiality, the order`s end user details are not send to the manufacturers. Only the part of the order that is relevant to quoting the price is communicated. Each manufacturer calculates an offer and transmits it to the assembler. The assembler collects the lowest quoted price in time, sends a mandate to the selected manufacturer

and informs end users about the assured delivery date. The manufacturer again authorizes a raw material distributor. For all raw material orders (big ones and small ones) the delivery of the raw material is coordinated and the material is inspected. The raw material that is not rejected due to defects is conveyed to the second layer and the components are manufactured. Thereafter, components are made available to the assembler (of the third layer) for assembling the complete products. The whole order is put on hold if at least one product fails the final testing. The rejected products are repaired and tested again. Fig. 1 describes a situation with three end-customer orders, four accepted orders and one final order.

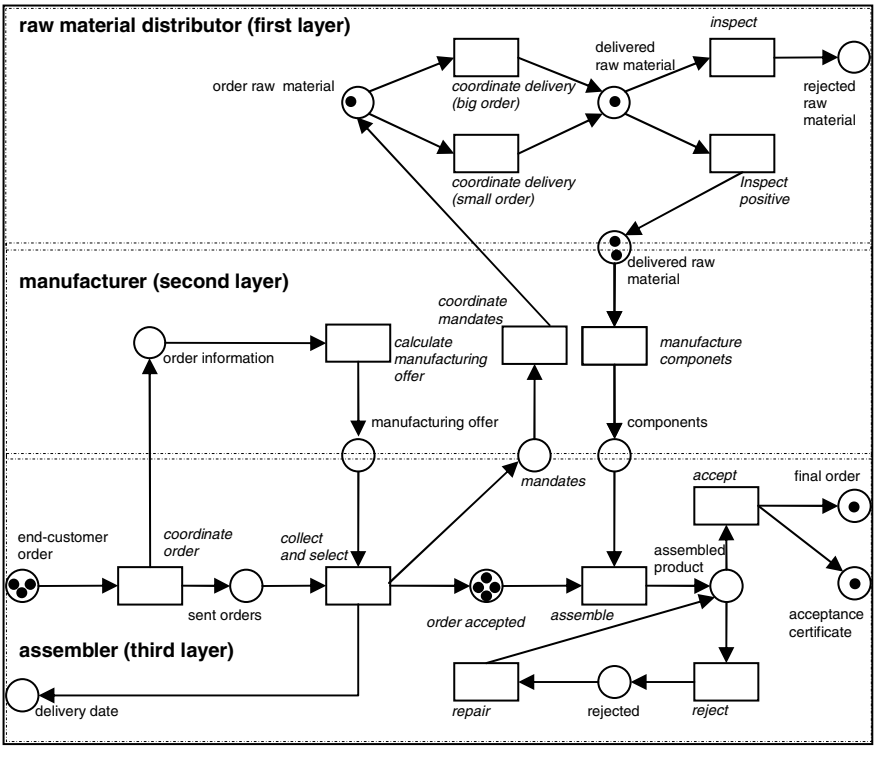


Fig. 1. Place/Transition net for an inter-organizational collaborative manufacturing process

P/T nets are generally suited for modeling interrelated production, logistics and information processes. Especially, process dynamics, concurrency, parallelism, and cycles of processes as well as asynchronous operations can be modeled in an explicit, appropriate and comprehensible manner. However, they do exhibit some shortcomings in regard to modeling the BPEL-based choreography of Web services. For modeling business processes and workflows with identifiable objects(e.g. data objects and physical objects such as customer orders, components) and to define complex hierarchical structure of objects XML nets, a variant of high-level Petri nets, have been proposed. In XML nets, the static components are typed by XML schema

diagrams, each of them representing an XML schema. Places can thus be regarded as containers for XML documents that are valid for the corresponding XML schema. The flow of XML documents is defined by occurrences of transitions, i.e., the activities taking place. So-called filter schemas label the adjacent arcs and thereby describe the documents that are relevant for the activity and the way these documents are manipulated. XML nets are informally defined as follows:

An XML net is a tuple XN= **<P, T, F, Ψ, IP, L, IT>** where (i) **<P, T, F>** is a Petri net. (ii) A function **IP** assigns to each place an XML schema. All admissible XML documents of the place's marking must be valid for the respective schema. (iii) **L** labels the arcs with a so-called filter schema that fits to the XML schema of the adjacent place. Filter schemas of arcs from places to transitions may have a manipulation filter and thus describe either read or insert operations whereas filter schemas of arcs from transitions to places must have a manipulation filter and describe delete operations. (iv) **IT** assigns to transitions a logical expression over a given structure Ψ that must evaluate to true in order to enable transition occurrences. (v) An initial marking assigns to each place of the XML net a set of valid XML documents.

The behavior of XML nets is defined by transition occurrences. If a transition occurs, it removes the matching documents or parts of the documents as specified by the arc label from the places in the transition's and inserts new documents into the places in the transition's post-set or new elements into the matching documents. For more details on XML nets we refer to [11], [12], [13].

XML nets and BPEL address quite different levels of business process modeling. While XML nets capture operational semantics and their control flow, BPEL includes additional implementation aspects such as involved Web services.

3 Petri Net Based Web Service Description

Basically, the behavior of a Web service is a partially ordered set of operations. XML nets make it particularly easy to model the flow of documents or data and to specify operational semantics of the control flow. Thus, the definition and description of Web services can be visualized by XML nets. Moreover, modeling organizational or inter-organizational collaborative business processes with XML nets, functionalities that will be provided by Web services, may be identified as a subnet of an XML net or can be described by the overall business process.

In the following, we define an XML service net **XSN** describing the logical behavior of a Web service with the following properties: (i) XSN has exactly one place **ps** with no incoming arcs. This place is the input place of the XML service net. An XML document (with a unique identifier) has to be assigned to this place as initial marking. (ii) XSN has exactly one place **pe** with no outgoing arcs. pe is the output place of the XML service net. An XML document inserted in this place cannot be deleted or manipulated by any transition of the XML net. Thus, an XML service net is an XML net with an input place variable (defined by XML elements) that invokes the Web service. In the output place we define the call for the return value. The output place provides information by sending messages that contain XML documents.

In contrast to the definition presented in [6] for a so-called service net that is a labeled P/T net, our approach focuses on high-level Petri nets.

Fig. 2 shows an XML service net describing the input and output of the transition "calculate manufacturing offer" (the visualization of XML documents in input and output places is abbreviated). The XML service net implements the functionality of calculating a manufacturing offer. The places *s* and *e* represent the input and the output place of the XML service net, respectively.

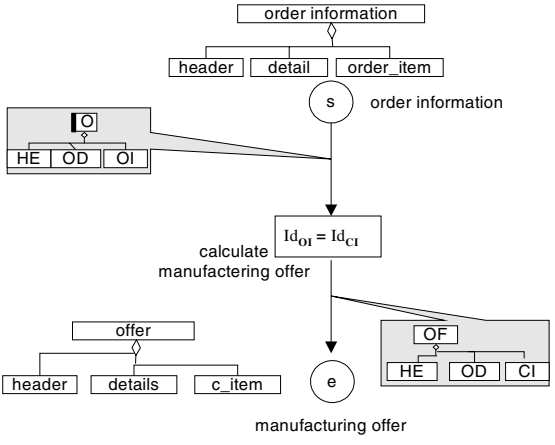


Fig. 2. XML service net for the calculation of manufacturing offer

We extend the above defined XML service nets with a definition for Petri net based Web service description to so-called Web service nets (WS nets). A Web service net is a tuple WS = < **N**, **D, L, U, CS, XSN** > with

N : name of the service and unique identifier,
D : service description,
L : server where the Web service net is located,
U : invocation of the Web service net,
CS : set of component services, and
XSN : XML service net that describes the Web service net

Accordingly, a basic (non-composite) Web service net is defined by a set of component services that contains only the Web service net itself, i.e. for WS = (N, D, L, U, CS, XSN) holds CS = {N}. Due to the fact that the XML service net does not only describe the control flow of the service, the description of the Web service net concerning behavior, functionality, and interfaces can be directly derived from the XML service net. The composition of Web service nets is similar to workflow services as defined in [14] and service nets as described in [6]. In the next sections we will propose an approach for deriving BPEL elements from Web service nets.

4 Deriving BPEL Elements from Web Service Nets

To realize B2B integration with Web services the utilization of Web service standards (WSDL [22], SOAP, UDDI) and methods for composition of Web services is required. BPEL provides an XML notation and semantics for describing the behavior of business processes based on Web services. In this sense, BPEL process definitions require the use of one or more WSDL services to enable the description of the behavior and interactions of a process instance relative to its partners and resources through Web service interfaces [1]. However, modern orchestration and choreography languages do not yet support analysis methods to verify that business processes meet certain requirements. XML nets facilitate modeling, analysis and execution of business processes and solve syntactical composition problems of distributed business processes. With the XML service net we describe control flows of Web services and derive the description of Web service nets concerning behavior, functionality, and interfaces. For choreography of Web service nets for example of several partners and their execution, we propose a novel approach to derive BPEL elements from Web service nets.

4.1 Overview of BPEL Elements

A BPEL business process document is defined by a partner specification (partnerLink), a set of variables for representing the state of a business process and activities describing the operational semantics of business processes. Partner represents both a consumer of a service provided by the business process and a provider of a service to the business process. BPEL concepts can be utilized to define two alternative types of processes: executable processes and abstract processes. Orchestration of processes is modeled by executable processes that are executable by an orchestration engine. Choreography can be defined by abstract processes.

For invoking a Web service, for manipulating data, for handling faults and for terminating a process, BPEL uses two types of activities. Basic activities perform simple operations and include several activities such as receive (waiting for a message from an external partner), reply (answering to a message of an external partner), invoke (invoking an operation of a Web service), assign (updating values of variables with new data), throw (generating a fault), terminate (stopping the entire service instance), wait (waiting for a certain time) and empty (to do nothing). Structured activities can be considered as a set of basic activities that impose an execution sequence for basic activities. There are several structured activities defined such as sequence (collection of activities to be performed sequentially), switch (selecting exactly one branch of activity from a set of choices), flow (specifying one or more activities to be performed concurrently), pick (blocking and waiting for a suitable message) and while (for defining loops). Furthermore, BPEL supports exception handling mechanisms provided by the construct scope. This construct allows defining a nested activity with its own fault handlers, event handlers, compensation handlers, data variables, and correlation sets. With fault handlers exceptions can be handled during the execution of its enclosing scope. Event handlers can occur by specified alarms or correspond with incoming messages. With the use of a compensation handler certain tasks, which happened during the execution of activities, can be rolled back. Compensation handler can be invoked by the

compensate activity. With the help of correlation sets a collection of properties shared by all messages in a correlation set can be defined. An important concept of BPEL are links that allow to form acyclic graphs. With links control dependencies between concurrent activities can be expressed.

4.2 Deriving BPEL Elements

The state of the Web service described by a Web service net can be NotInstantiated, Ready, Running, Suspended or Completed (similar to the ones defined in [19]). When a state is in the Completed state the token of the Web service net is in the corresponding output place.

We distinguish between four different types of flow structure in Web service nets. In a sequence the flow of activities is predetermined. By the use of alternative one branch of a set of choices can be selected. The flow structure parallelism enables to model parallel branching and after the parallel execution the activities have to be joined. In Fig. 3 the different flow structures of Web service nets are modeled.

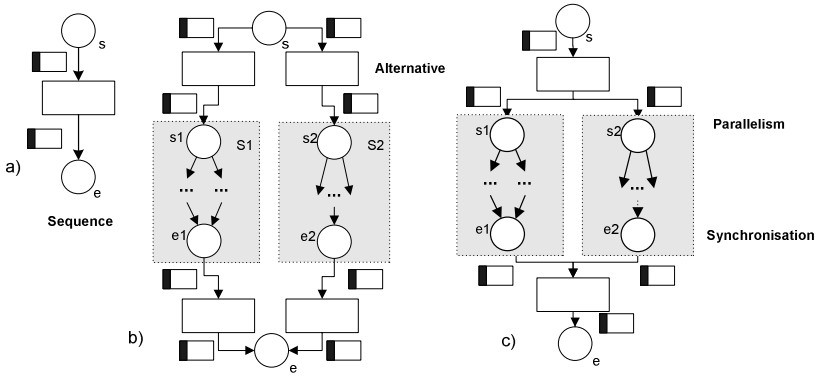


Fig. 3. Flow structures of Web service nets

To initiate a Web service BPEL utilizes the activity invoke to call the intended Web service net. The invoke construct allows the Web service net to invoke a one-way or request/response operation on a portType offered by a partner. This construct is specified by a tuple iv = < **Ni, pL, pT, op, i, o** > where:

Ni : name of the Web service net,
pL : corresponding partnerLink,
pT : portType,
op : operation to invoke the Web service net,
i : input variable,
o : output variable

A BPEL assign activity provides a mechanism to perform simple data manipulation to initialize the input variable which is passed to the Web service net, using XPath expressions. The assign construct can be used to update the values of

variables with new data. An assign construct can contain any number of elementary assignments and is specified by the tuple a = < **Na, f, t** > where:

Na : name of the Web service,
f : source data,
t : variable or element part as the destination for the data

When the variable is defined using XML Schema simple type or element, the part attribute must not be used.

Operations are modeled by transitions and the state of the Web service net is modeled by corresponding input and output places. The arcs between places and transitions represent causal relationships. In Fig. 5a a Web service net is modeled with a sequence from which the BPEL construct invoke is derived with the following instances[1]:

Ni : calculateManufacturingOrder,
pL : client,
pT : CreditFlowCallback,
op : compareIDs,
i : orderInformation,
o : manufacturingOffer

In Fig. 4b the Web service net initiates the activities invoke and assign to update the price. The manipulation is handled in the transition "calculate manufacturing offer". The invoke initiation is described in Fig. 4a and the assign activity concerns the following initiation:

Na : updatePrice,
f : price,
t : price´

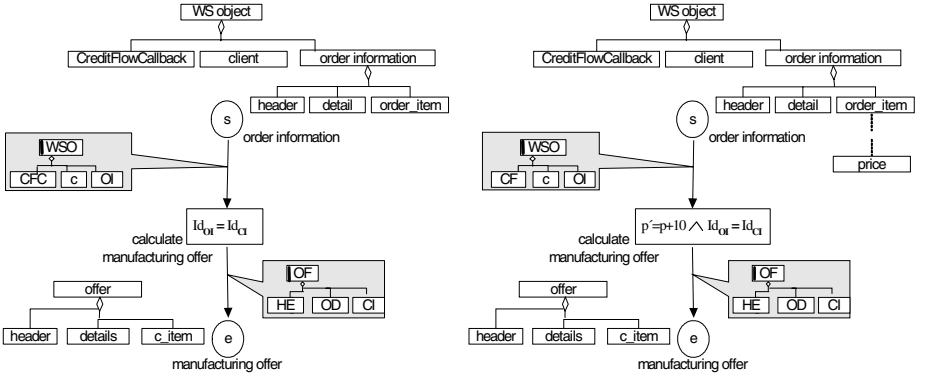


Fig. 4. a) Initiating invoke activity and **b)** Initiating invoke and assign activities

[1] We distinguish WS object from a WSDL by the definition of a WSDL as a tuple WS=<pL, pT, T, M> where T is a type (list of types participating in this BPEL process) and M is a MessageType.

The BPEL syntax for the Web service net in Fig. 4b is as follows:

```
<sequence>
<invoke name="calculateManufacturingOffer"
                partnerLink="client"
                portType="tns:CreditFlowCallback"
                operation="compareIDs"
                inputVariable="orderInformation"
                outputVariable="manufacturingOffer"/>
  <assign name="updatePrice">
  <copy>
  <from variable="price" part="orderInformation"/>
  <to variable="price`"/>
  </copy>
  </assign>
</sequence>
```

Invoking an operation on a Web service net that is provided by partners is a basic activity. This operation can be a synchronous request/response or an asynchronous one-way operation which is modeled in Fig. 5. BPEL uses the same basic syntax for both variants with some additional options for the synchronous case. A synchronous BPEL process blocks the client (the one which is using the process) until the process finishes and returns a result to the client. An asynchronous process does not block the client. A request/response pattern is implemented by a receive/reply pair. The receive construct allows the business process to execute a blocking wait for a matching message to arrive. The receive activity is a tuple re = < **Nre, pL, pT, op, v** > where:

Nre : name of the Web service net,
pL : partnerLink,
pT : portType,
op : operation to invoke for the Web service net,
v : variable as the reception of the data

Variables provide the means for holding messages that constitute the state of a business process.

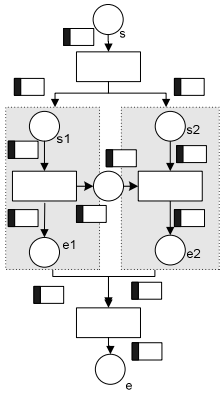


Fig. 5. Invoking an asynchronous BPEL process from within a Web service net

The reply construct allows the business process to send a message in reply to a message that was received through a receive. The reply activity is a tuple rp = < **Nrp, pL, pT, op** > where:

Nrp : name of the Web service net,
pL : partnerLink,
pT : portType,
op : operation

The partnerLink created for the client interface includes two roles, myRole and partnerRole assignment. An asynchronous BPEL process typically has two roles for the client interface: one for the flow itself, which exposes an input operation, and one for the client.

Following our approach presented in this paper we can derive further BPEL constructs from Web service nets such as pick (= parallelism), while (= loop), switch (= alternative), wait (= synchronization), terminate (= deadlock), scope (= refinement), compensate and scope (= repair and = alert).

5 Summary and Conclusion

A major issue of collaborative business process management is the appropriate representation of all relevant information of manual and executable parts of business processes. In this paper we have presented an XML based approach for process model driven deduction of BPEL for management of inter-organizational collaborative business processes which addresses this essential issue. By employing XML nets, process schema based analysis and execution of business processes can be enhanced. XML nets combine the benefits of modeling functionalities that shall be provided by Web services, their coordination and derivation of BPEL elements and the advantages of the well founded and established Petri nets.

Moreover, XML nets facilitate an integrated approach to inter-organizational collaborative business process management by using a consistent and comprehensible method throughout all relevant phases. Accordingly, choreography of the whole intra-organizational business process based on Web services is feasible.

A prototypical software module currently under development will be integrated to an existing software tool suite for modeling, analyzing and executing XML nets [11]. The software module constitutes the basis for further development, implementation, and application of the proposed method. For example, the fragment based modeling of BPEL code will be extended in order to support the flexible derivation of new, user defined fragments.

References

1. Alonso, G.; Casati, F.; Kuno, H.; Machiraju, V.: Web Services, 2004, Springer Verlag, Heidelberg.
2. Arkin, A.: Business Process Modeling Language. http://www.bpmi.org/bpml.esp.
3. Choi, I.; Song, M.; Park, C.; Park, N.: An XML-based process definition language for integrated process management, in: *Computers in Industry*, 50, 2003, pp. 85-102.

4. Dong, M.; Chen, F.: Process modeling and analysis of manufacturing supply chain networks using object oriented Petri nets, in: *Robotics and Computer Integrated Manufacturing*, 17, 2001, pp. 121-129.
5. Curbera, F.; Goland, Y., Klein, J., Leymann, F., Roller, D., S. Thatte, Weerawarana, S. Business Process Execution Language for Web Services. http://www.ibm.com/ developerworks/library/ws-bpel/.
6. Hamadi, R.; Benatallah, B.: A Petri Net-based Model for Web Service Composition. In: Schewe, K.-D.; Zhou, X. (eds.): Database Technologies, Proc. 14th Australasian Database Conference, (2003), pp. 191-200.
7. Harel, D.: Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8(3), 2001, pp. 231-274.
8. Hinz, S.; Schmidt, K.; Stahl, C: Transforming BPEL to Petri Nets. In Proceedings of the Third International Conference on Business Process Management (BPM 2005), Nancy, France, 2005, (to appear)
9. Jensen, K.: Coloured Petri Nets. Vol 1: Basic Concepts. Berlin et al., 1992.
10. Karagiannis, D., Juninger, S., Strobl, R: "Introduction to Business Process Management Systems Concepts". In: Scholz-Rieter, Stickel (eds): *Business Process modeling*, 1996, pp. 81-106.
11. Lenz, K., Mevius, M., Oberweis, A.: Process-oriented Business Performance Management with Petri Nets, in: Cheung, W.; Hsu, J. (eds.): Proc. 2nd IEEE Conference on e-Technology, e-Commerce and e-Services, Hong Kong, 2005, pp. 89-92.
12. Lenz, K.; Mandaric, A.; Oberweis, A.: Modeling Processes for Managing Reputation Information - A Petri Net Approach, in: Cordeiro, J.; Filipe, J. (Hrsg.): Proceedings of the 1st International Workshop on Computer Supported Activity Coordination (CSAC 2004), Porto/Portugal, April 2004, pp. 136-148
13. Lenz, K., Oberweis, A.: Inter-organizational Business Process Management with XML Nets, in: Ehrig, H., Reisig, W., Rozenberg, G., Weber, H. (eds.): *Advances in Petri Nets*. Lecture Notes in Computer Science, Vol. 2472, Springer-Verlag, Berlin Heidelberg New York, 2003, pp. 243-263.
14. Lenz, K.; Oberweis, A.: Workflow Services: A Petri Net-Based Approach to Web Services, in: Proceedings of Int. Symposium on Leveraging Applications of Formal Methods, Paphos/Cyprus, November 2004, pp. 35-42.
15. Mantell, K.: From UML to BPEL, Model Driven Architecture in a Web services world, IBM, http://www-128.ibm.com/developerworks/webservices/library/ws-uml2bpel/.
16. Reisig, W.; Rozenberg, G. (eds.): Lectures on Petri Nets I: Basic Models. Lecture Notes in Computer Science, Vol. 1491, Springer-Verlag, Berlin, 1998.
17. Reisig, W.: Place/Transition Systems. In: Brauer, W., Reisig, W., Rozenberg, G. (eds.): *Advances in Petri Nets. Part I*. Lecture Notes in Computer Science, Vol. 254, Springer-Verlag, Berlin Heidelberg New York, 1987, pp. 117-141.
18. Scheer, A.-W.: *ARIS – Business Process Modeling*, 2nd ed., Berlin et al., 1999.
19. Schuster, H.; Georgakopoulos, D.; Cichocki, A.; Baker. D.: Modeling and Composing Service-based and Reference Process-based Multi-enterprise Processes, in: Proceeding of the 12th Conference on Advanced Information Systems Engineering, (2000), Stockholm
20. Zimmermann, A., Freiheit, J., Huck, A.: A Petri net based design engine for manufacturing systems. *International Journal of Production Research*, 39(2), 2001, pp. 225-253.
21. Wohed, P.; van der Aalst, W.M.P.; Dumas, M.; ter Hofstede, A.H.M.: Analysis of Web Services Composition Languages: The Case of BPEL4WS. Lecture Notes in Computer Science, Vol. 2813. Springer-Verlag, Heidelberg (2003), 200-215.
22. W3C. Web Services Description Language (WSDL) Version 2.0 PART 1: Core Language. W3C Working Draft (2005), http://www.w3.org/TR/wsdl20/.
23. W3C. Web Service Architecture Requirement, W3C Working Group Note, (2004), http://www.w3.org/TR/wsa-reqs/.

From Inter-organizational Workflows to Process Execution: Generating BPEL from WS-CDL

Jan Mendling[1] and Michael Hafner[2]

[1] Dept. of Information Systems and New Media,
Vienna University of Economics and Business Administration,
WU Wien, Austria
jan.mendling@wu-wien.ac.at
[2] Quality Engineering Research Group, Institut für Informatik,
Universität Innsbruck, Austria
m.hafner@uibk.ac.at

Abstract. The Web Service Choreography Description Language (WS-CDL) is a specification for describing multi party collaboration based on Web Services from a global point of view. WS-CDL is designed to be used in conjunction with the Web Services Business Process Execution Language (WS-BPEL or BPEL). Up to now, work on conceptual mappings between both languages is missing. This paper closes this gap by showing how BPEL process definitions of parties involved in a choreography can be derived from the global WS-CDL model. We have implemented a prototype of the mappings as a proof of concept. The automatic transformation leverages the quality of software components interacting in the choreography as advocated in the Model Driven Architecture concept.

1 Introduction

The exchange of structured information between business partners is a crucial means to facilitate coordinated production of goods and services. The increasing use of Web Services for the implementation of inter-organizational scenarios underlines the need for a choreography description language. Choreography languages (or coordination protocols [1]) are a means to define the rules of a collaboration between parties without revealing internal operations. They allow to specify when which information is sent to which party and which options are available to continue the interaction.

Several specifications have proposed choreography languages, for an overview and their relationship to the Web Services stack see e.g. [1,2]. The Web Service Choreography Description Language (WS-CDL) [3] as the latest proposal is based on a meta model and an XML syntax. It is expected to be used in conjunction with the Web Service Business Process Execution Language (WS-BPEL) [4]. There are two application scenarios in this context: first, business entities may agree on a specific choreography defined in WS-CDL in order to achieve a common goal. This WS-CDL choreography is then used to generate WS-BPEL process stubs for each party. In this case, the WS-CDL choreography may be regarded as a global contract to which all parties commit. Second, a business may

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 506–515, 2005.
© Springer-Verlag Berlin Heidelberg 2005

want to publish its processes' interface to business partners. In this scenario, a choreography description of the internal process has to be generated.

WS-CDL has been criticized for the insufficient separation of meta-model and syntax, the limited support for certain use case categories and its lack of formal grounding [5]. This was taken as a motivation to identify service interaction patterns [6] that might build the foundation of a new choreography language. Besides, it is not clear whether all WS-CDL concepts can be mapped to WS-BPEL [5]. Our paper discusses mappings between WS-CDL and WS-BPEL. The contribution is twofold. First, the mappings can be used to generate WS-BPEL stubs from WS-CDL choreographies and WS-CDL descriptions from WS-BPEL processes, which leverages the re-use of design artifacts as advocated e.g. in the Model Driven Architecture (MDA) approach. Second, the definition of mappings yields insight into potential incompatibilities of both languages. We implemented the mapping in XSLT transformation programs as a proof of concept.

The rest of the paper proceeds as follows. In Section 2 we give an example of an inter-organizational workflow based on a real use case. Based on this example, we then present the main concepts of WS-CDL and BPEL in Sections 3 and 4. Section 5 defines the mappings between WS-CDL and BPEL, and we discuss how BPEL can be generated from WS-CDL. We give an overview of related research in the area of Web Service choreography in Section 6. Finally, Section 7 closes the paper with a conclusion and gives an outlook on future research.

2 Example of an Interorganizational Process

We use an example to illustrate various aspects of the relationship between an externally observable choreography and related internal orchestrations of the collaborating partners' nodes. The example captures an inter-organizational process in e-government. It is drawn from a case that was elaborated within the project SECTINO [7]. The project's vision was defined as the development of a framework supporting the systematic realization of e-government related workflows.

The workflow-scenario "Municipal Tax Collection" describes a Web Services based interaction between three participants: a tax-payer (the Client), a business agent (the Tax Advisor) and a public service provider (the Municipality). In Austria, wages paid to employees of an enterprise are subject to the municipal tax. According to the traditional process, corporations have to send an annual statement via their tax advisor to the municipality. The latter is responsible for collecting the tax. It checks the declaration of the annual statement, calculates the tax duties and returns a tax assessment notice to the tax advisor.

In our case the stakeholders in this public administration process agreed to implement a new online service, which offers citizens and companies to submit their annual tax statements via internet. Due to various legal considerations, the process has to be realized in a peer-to-peer fashion and should ultimately integrate security requirements like integrity, confidentiality and non-repudiation.

Figure 1 shows the choreography model as a UML Activity Diagram. It describes the collaboration of the three services in terms of the interactions in which

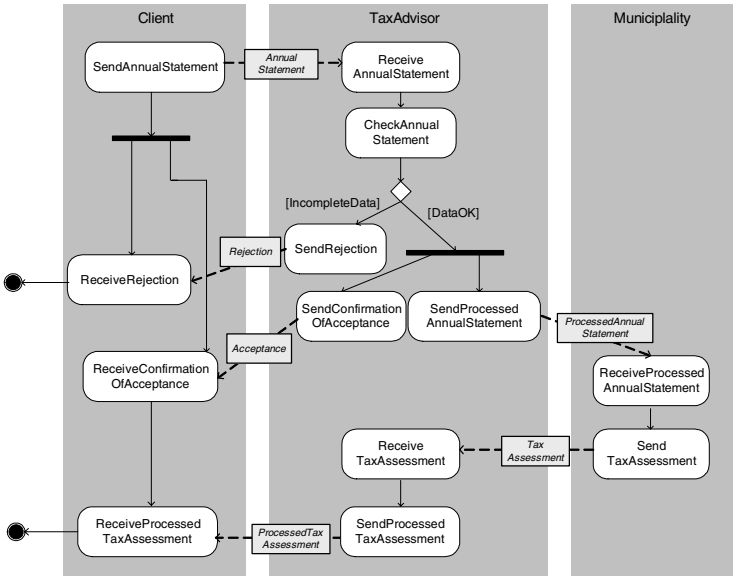


Fig. 1. Example of a choreography for municipal tax collection

the participating parties engage. Model information is confined to the "observable behavior", corresponding to the message flow between the participants, the interaction logic and the control flow between the elementary actions.

3 An Illustrative Overview of WS-CDL

WS-CDL [3] is a declarative XML-language for the specification of collaboration protocols based on Web Services. It provides a global public view on participants collaborating in a peer-to-peer fashion by offering distributed Web Services in order to achieve a common business goal. The protocol describes their observable behavior through the order of the messages they exchange as well as the operations they offer. Taking our example "Municipal Tax Collection", listing 1 shows two parts of a WS-CDL document: the *package information* and the *choreography definition*. We sketch the main concepts only, for details refer [3].

Package information: The package element is the root of every chorography definition and contains `informationType` definitions for the messages and variables, e.g., the document "annualStatement" sent from Client to Tax Advisor (lines 5-7) and process instance correlation data (lines 2-4). These data types are used within the choreography definition part. A `roleType` represents an actor of the collaboration like the "ServiceProviderRole" (lines 8-11). This element associates operation names and their WSDL interfaces via the `behavior` element. For example, the "ServiceProviderRole" is expected to implement a "ReceiveAnnualStatement" operation, which is specified in the corresponding WSDL file. The `relationshipType` element pairs two roles and optionally a subset of the be-

havior they exhibit: e.g., the `relationshipType` "ClientTaxAdvisor" associates a "ClientRole" to a "ServiceProviderRole" which is one of two roles a "TaxAdvisor" is expected to implement (lines 16-19). A `participantType` represents a logical grouping of roles. For example, the `participantType` "TaxAdvisor" implements two roles: on the one hand the "ServiceProviderRole" and on the other hand "ServiceRequesterRole" (lines 20-23). A `channelType` indicates the role the receiver of a message is playing and optionally which behaviour he is expected to implement: the `channelType` "SubmitAnnualStatementChannel" specifies a return channel for responses to a document submission (lines 24-26.). Finally, every package contains one or more choreography definitions (line 37).

Choreography Definition: The core of a collaboration is defined by the element `choreography`, which specifies a set of peer-to-peer interactions. A package can contain one or more choreographies, one being the root choreography (lines 1-2). A choreography first specifies relationships. The example shows two relationships: one between a Tax Advisor and his Client, and one between the Tax Advisor and a Municipality (lines 3-4). In a second step, the variables are declared, e.g., the variable "AS" of type "annualStatement" will only be used by the "ClientRole" and the "ServiceProviderRole". The `interaction` element is the building block of communication. It participates in a `relationshipType` and specifies the direction of the message flow: the message flows from the sender (specified as `FromRole`) to the receiver (`ToRole`) if the `action` attribute of the `exchange` element is set to `request`. The `exchange` element also captures the

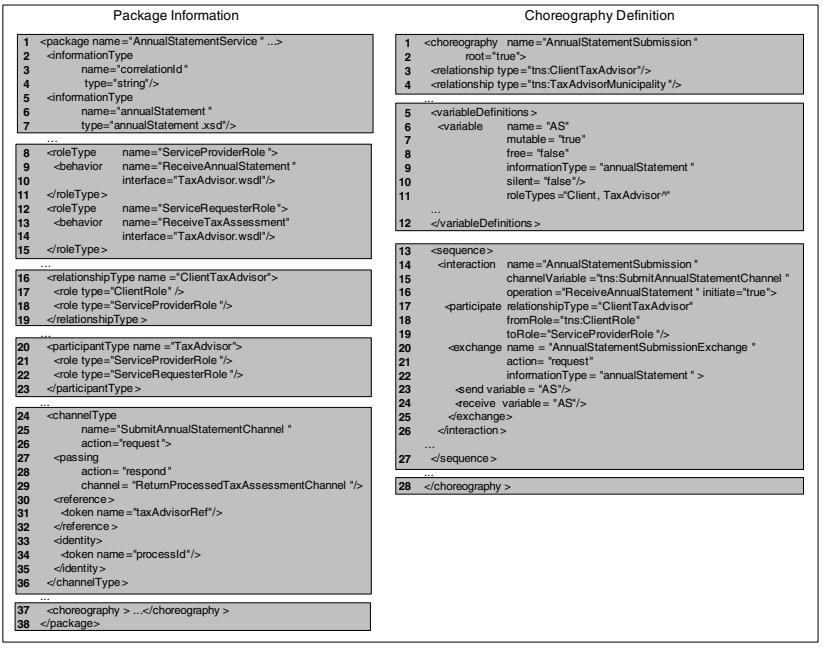


Fig. 2. WS-CDL Listings

name of the operation associated to this interaction. Interaction elements can be nested within control-flow activities (e.g., `sequence`, line 13-27).

4 BPEL Implementation of the Tax Advisor

BPEL is an XML-based language for the composition of executable business processes based on Web Services. The specification of BPEL can be found in [4].

Listing 2 shows a BPEL process for one of the choreography participants. In order to implement his part of the choreography, the Tax Advisor orchestrates the sequence of service interactions through a BPEL process called TaxAdvisorProcess. The service composition is offered through an interface according to the choreography specification. A `partnerLink` defines internal and external parties (`myRole` and `partnerRole`) that interact with the process instance and the `portTypes` that need to be implemented (see lines 2-6). A `partnerLinkType` is a BPEL extension, which is used in the WSDL definition. It defines two roles of a bilateral message exchange and their `portTypes`. A `partner` element (lines 7-9) can be used to group `partnerLinks`. `Variables` (lines 12-14) describe the message types used in a BPEL process. A `variable` is identified by a unique name and is associated with a message type. Variables store received messages and hold messages to be sent to other parties. A BPEL process describes the execution order of Web Services operations via basic and structured activities. A basic activity is either a message exchange between Web Services or a local operation of a BPEL engine. The example illustrates a `receive` activity (lines 17-25). The activity blocks the process until a matching message arrives. Invocations of remote web service operations are modelled as `invoke` activities. Lines 26-32 illustrate an asynchronous one-way invocation. Synchronous request/response interaction can be expressed by including an additional output variable to store the response. Control flow logic of a BPEL process is defined via structured activities. In our example we use `sequence` for sequential execution. A `correlationSet` describes parts of messages which are unique for a process instance. Aliases to these message parts are called properties.

BPEL Process Definition

```
1  <process  name="TaxAdvisorProcess" ...>
2  <partnerLinks>
3    <partnerLink   name="AnnualStatementSubmission"
4          myRole="ServiceProviderRole" partnerRole="ClientRole"/>
5          partnerLinkType="AnnualStatementSubmission"
   ...
6  </partnerLinks>
7  <partners>
8    <partner name="ClientRole">
9    </partner>
10   <partner name="ServiceProviderRole">
   ...
11 </partners>
12 <variables>
13   <variable name="AS"  messageType="annualStatement"/>
   ....
14 </variables>
```

```
15 <sequence name="1st level">
16   <receive        name="ReceiveAnnualStatement"
17                   partnerLink="AnnualStatementSubmission"
18                   portType="ReceiveAnnualStatementPT"
19                   operation="ReceiveAnnualStatement"
20         variable="AS" createInstance="yes">
21     <correlations>
22       <correlation set="tax:processId" initiate="yes"/>
23     </correlations>
24   </receive>
25   <invoke name="CheckAnnualStatementLocal"
26         partnerLink="Local" operation="CheckAnnualStatement"
27         inputVariable="CheckASLocalVar">
28     <correlations>
29       <correlation set="tax:processId" initiate="yes"/>
30     </correlations>
31   </invoke>
   ...
32   </sequence>
33 </process>
```

Fig. 3. BPEL Listing

5 Generating BPEL Stubs from WS-CDL

While the previous sections illustrated how BPEL processes can be modelled in correspondence to a given WS-CDL choreography, this section presents transformation rules from WS-CDL to BPEL. We implemented a transformation program that also includes mapping rules that are not presented here due to space limitations as a proof-of-comcept. [1] We use the namespace prefixes cdl: and bpel: to indicate to which specifications the concepts belong.[2]

Generally, one WS-CDL document maps to one or more `partnerLinkTypes` (each representing a bilateral communication relationship), multiple `property` and `propertyAlias` definitions related to WSDL interfaces, and at least two BPEL processes for each party involved in the choreography.

PartnerLinkTypes: Web Service interactions in a BPEL process rely on the availability of so-called `bpel:partnerLinkTypes`. A `bpel:partnerLinkType` defines the interaction of two parties by giving two related `bpel:role` elements and the `bpel:portType` that implements the role. This concept is very similar to the notion of a `cdl:relationshipType`. Accordingly, one `cdl:relationshipType` maps to one `bpel:partnerLinkType` and the `bpel:role` with its `bpel:portType` is generated from the referenced `cdl:roleType` declaration.

Properties: In BPEL properties play an important role for the correlation of messages and process instances. The `bpel:property` element defines an element that is unique for the process instance and which can be used for correlation. The related `bpel:propertyAlias` element specifies the XPath query to retrieve the `bpel:property` from a message. In WS-CDL `cdl:token` and `cdl:tokenLocator` elements represent the same concept. As only some property declarations are relevant for a specific party involved in the choreography, there needs to be a filter mechanism. We generate separate property files for each `cdl:roleType` including only those `bpel:properties` that are relevant for a party. The WS-CDL does not impose tokens to be defined, because e.g. a simple stateless request-response choreography does not need correlation. Accordingly, it is possible that no property files are generated from the WS-CDL choreography.

BPEL Process: For each party a separate BPEL stub is generated. A party is either a `cdl:participant` that bundles several cdl:roleTypes or a `cdl:roleType` that is not subordinated to a `cdl:participant`. For both the relevant information to be included in the BPEL files is identified via the `cdl:roleType` or the `cdl:roleType` elements referenced in the `cdl:participant` element.

A party's BPEL process includes declaration blocks with partner links, variables, and correlation sets – information needed by the activities defining the process. The `bpel:partnerLinks` block references the `bpel:partnerLinkType`

[1] The wscdl2bpel.xslt program is available from http://wi.wu-wien.ac.at/~mendling. It uses the XALAN extensions to generate multiple output files. For details see http://xml.apache.org/xalan-j.

[2] Being aware that there are separate schemas for BPEL processes, partnerLinkTypes, and properties, we use the same bpel: prefix for all the three for better readability.

files. It indicates the party's role in the process via the `bpel:myRole` attribute. The `bpel:variables` are generated from the `cdl:variableDefinitions` and from their references to `cdl:informationType` elements. Variables relevant to a party can be identified via the `cdl:roleTypes` attribute of each variable. `bpel:correlationSets` can be derived from the `cdl:channelType` elements: each channel element yields a `bpel:correlationSet` named after the channel and including the `cdl:token` element of the channel's `cdl:identity` element. The derived correlation sets are only included in those BPEL processes of parties using the respective channel in their interactions.

BPEL control flow is defined via scopes, structured and basic activities, both the first allowing to nest other activities. WS-CDL uses a similar concept: so-called work units can be related to scopes, ordering structures to structured activities, and WS-CDL basic activities to BPEL basis activities. In the following we describe the WS-CDL activities and how they map to BPEL.

- `cdl:workunit`: The `cdl:workunit` is related to the `bpel:scope` concept in the sense that it defines a context for consistent execution. Yet, its attributes have a much more direct impact on control flow than the `bpel:scope`. The `cdl:workunit` unifies the concepts of a loop (`cdl:repeat`), of a data event (`cdl:guard`), and a wait (`cdl:block`). The guard and the block are interrelated. If `cdl:block` is true, then the choreography waits for the guard condition to become true before progressing. If set to false, the `cdl:workunit` is skipped. When the `cdl:repeat` condition is true the workunit is considered again for execution depending on the guard. In BPEL the `cdl:block=false` case maps to a `bpel:switch` executing the nested activities if the guard condition is true, otherwise progressing with the next activity subsequent to the `cdl:workunit`. The `cdl:block=true` case is hard to map as BPEL does not know events like "variable becomes available". We propose to use a `bpel:receive` in this case, because it blocks until a message is received and written to a `bpel:variable`. The BPEL engineer needs to add information from where the message is to be received. Finally, a `cdl:repeat` condition is mapped to a `bpel:condition` of a `bpel:while` loop. Note that the `bpel:while` is executed until the `bpel:condition` becomes true, and the `cdl:repeat` indicates repetition as long as the condition is still true.
- `cdl:sequence`: In general the `cdl:sequence` of the global model maps to a `bpel:sequence` of the local model. Yet, if the respective party is involved only in one or in none of the child activities of the `cdl:sequence`, then no local sequences needs to be generated.
- `cdl:parallel`: The `cdl:parallel` maps to a `bpel:flow` element in the local model. Similar to the sequence, if the respective party is involved in zero or one of the parallel branches, then the `bpel:flow` element can be omitted.
- `cdl:choice`: In general the `cdl:choice` of the global model maps to a `bpel:case` nested in a `bpel:switch` element. In this case the `bpel:switch` can only be left out if the party is not involved in any of the `cdl:choice` nested activities. If the party is involved in one nested activity a `bpel:case` for this activity has to be generated and a `bpel:case` including a `bpel:empty`

activity. Note that the `bpel:conditions` of the cases need to be specified manually by the engineer of the BPEL process.

- `cdl:interaction`: Each `cdl:exchange` of an interaction maps to a web service activity in BPEL (see Figure 4). In this context four cases have to be distinguished depending on the value of the `cdl:action` and whether the current party is mentioned in the `cdl:toRole` or `cdl:fromRole` attribute. In case of a request action a `bpel:invoke` is generated for the party of the `cdl:fromRole` and a `bpel:receive` for the party of the `cdl:toRole`. In case of a response action it is the other way around. If there is a `cdl:timeout` specified, a `bpel:pick` with a concurrent time event to the message receipt has to be specified in place of the `bpel:receive`.
- `cdl:perform`: This activity is not directly mapped to BPEL, but all nested activities of the referenced choreography are transformed and included.
- `cdl:assign`: This activity maps to a `bpel:assign` activity for the party mentioned in the `cdl:roleType` attribute.
- `cdl:silentAction`: This activity indicates that a party must perform some action that is not revealed in the global model. We propose to map it to a `bpel:sequence` with a nested `bpel:empty` activity and a `name` attribute set to "silent action". The engineer of the BPEL process will then have to specify these silent activities before deployment.
- `cdl:noAction`: This activity maps to a `bpel:empty` activity for the party mentioned in the `cdl:roleType` attribute.
- `cdl:finalize`: Finalizing activities can only be started after successful completion of a choreography. In this sense, they are related to the concept of a `bpel:compensationHandler`. Yet, they also involve communication to confirm the completion to other parties. Therefore, they cannot always be mapped to a `bpel:compensationHandler` because the only purpose of the latter is to undo successfully completed actions. Accordingly, we propose to append finalizing activities to the BPEL process of the parties involved.

With this transformation algorithm, BPEL processes can be generated almost automatically for all parties. Still, a BPEL engineer has to add implementation specific information including conditions for cases of a `bpel:switch` or activities that have been defined as silent activities in the choreography model.

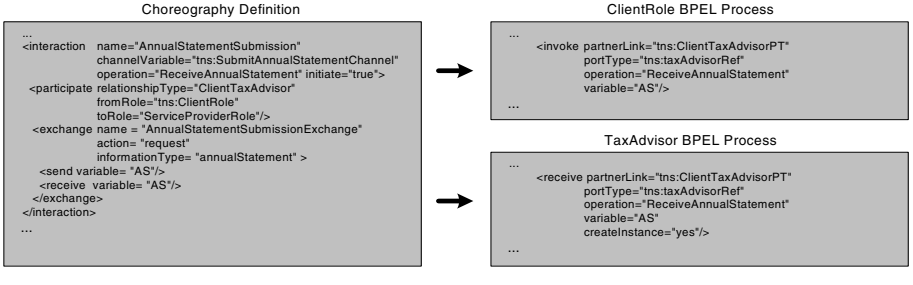


Fig. 4. Transformation of the `cdl:interaction` element

6 Related Research

There is a big community currently working on general issues related to inter-organizational workflow management systems (e.g. [8,9]). A number of contributions discuss standards for specifying service choreographies (e.g., [5]) and propose formal foundations (e.g., [6,8,10]). We do not aim to contribute a novel approach to this field or to develop a new standard. Instead, focusing on Web Services technology, we use existing technology and standards to realize our vision of Model Driven Architecture in the context of inter-organizational workflows. In [7] we present a framework that extends the concepts of Model Driven Architecture to Model Driven Security for inter-organizational workflows. A model driven approach close to the idea of our framework can be found in [11]. It introduces the concept of Model Driven Security for a software development process by integrating security requirements through system models and supports the generation of security infrastructures. But this approach focuses exclusively on business logic in the context of J2EE and .NET, whereas we concentrate on inter-organizational workflow management based on Web Services.

[12] describes an implementation, where a local workflow is modeled in a case-tool, exported via XMI-files to a development environment, and automatically translated into executable code for a BPEL engine based on Web Services. In contrast to this approach, we move up one layer of abstraction and start modeling at the choreography level. This is comparable to the approach promoted in [13] where the authors start with UMM to generate BPEL. In [14] we propose an approach for integrating security into the development cycle, starting at the choreography level and show how the requirements map through different levels of abstraction. In [15] we link abstract domain-level models to their technical implementation and show how requirements are realized through security components in a target architecture based on Web Services standards.

7 Conclusions and Future Work

This paper showed how BPEL process definitions for parties involved in a choreography can be derived from a global WS-CDL model. We have implemented a prototype of the mappings as a proof of concept. The automation offers a speed-up of the engineering process. Additionally, the automatic generation of BPEL stubs minimizes the risk of inconsistent process implementations by the parties. For future research, we aim at the comprehensive mapping of standard WS-CDL semantics to executable WS-BPEL process stubs. This includes the mapping of WS-CDL specific concepts (e.g. passing channel variable) but also the analysis of the potential for integrating more complex workflow semantics (e.g. transactional security) at the choreography level. Furthermore, we push the step-wise implementation of our MDA approach by developing custom modeling tools and plug-ins as well as components of the reference architecture.

References

1. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services - Concepts, Architectures and Applications. Springer Verlag, Berlin et al. (2003)
2. Mendling, J., zur Muehlen, M., Price, A.: Standards for Workflow Definition and Execution. In: Process Aware Information Systems: Bridging People and Software Through Process Technology. Wiley Publishing (2005)
3. Kavantzas, N., Burdett, D., Ritzinger, G., Fletcher, T., Lafon, Y.: Web Services Choreography Description Language Version 1.0. W3C Working Draft 17 December 2004, World Wide Web Consortium (2004)
4. Andrews, T., et al.: Business Process Execution Language for Web Services, Version 1.1. , BEA, IBM, Microsoft, SAP, Siebel (2003)
5. Barros, A., Dumas, M., Oaks, P.: A Critical Overview of the Web Service Choreography Description Language (WS-CDL). BPTrends Newsletter **3** (2005)
6. Barros, A., Dumas, M., ter Hofstede, A.: Service Interaction Patterns: Towards a Reference Framework for Service-based Business Process Interconnection. Technical Report FIT-TR-2005-02, Queensland University of Technology (2005)
7. Breu, R., Hafner, M., Weber, B., Novak, A.: Model driven security for inter-organizational workflows in e-government. In Böhlen, M.H., Gamper, J., Polasek, W., Wimmer, M., eds.: TCGOV. Volume 3416 of LNCS. (2005) 122–133
8. van der Aalst, W.: Loosely coupled interorganizational workflows: Modeling and analyzing workflows crossing organizational boundaries. Information and Management **37** (2000) 67–75
9. Grefen, P.W.P.J., Aberer, K., Ludwig, H., Hoffner, Y.: Crossflow: Cross-organizational workflow management for service outsourcing in dynamic virtual enterprises. IEEE Data Eng. Bull. **24** (2001) 52–57
10. Brogi, A., Canal, C., Pimentel, E., Vallecillo, A.: Formalizing web service choreographies. Electr. Notes Theor. Comput. Sci. **105** (2004) 73–94
11. Lodderstedt, T., Basin, D.A., Doser, J.: SecureUML: A UML-Based Modeling Language for Model-Driven Security. In Jézéquel, J.M., Hußmann, H., Cook, S., eds.: UML. Volume 2460 of Lecture Notes in Computer Science. (2002) 426–441
12. Gardner, T.: UML Modelling of Automated Business Processes with a Mapping to BPEL4WS. In: Proceedings of the First European Workshop on Object Orientation and Web Services at ECOOP 2003. (2003)
13. Hofreiter, B., Huemer, C.: Transforming umm business collaboration models to bpel. In Meersman, R., Tari, Z., Corsaro, A., eds.: OTM Workshops. Volume 3292 of Lecture Notes in Computer Science., Springer (2004) 507–519
14. Hafner, M., Breu, R., Breu, M., Nowak, A.: Modeling inter-organizational workflow security in a peer-to-peer environment. In: Proceedings of ICWS. (2005)
15. Hafner, M., Breu, R., Breu, M.: A security architecture for inter-organizational workflows: Putting security standards for web services together. In Chen, C.S., et al., J.F., eds.: Proceedings ICEIS. (2005)

PIT-P2M: ProjectIT Process and Project Metamodel

Paula Ventura Martins [1] and Alberto Rodrigues da Silva [2]

[1] INESC-ID, CSI/Universidade do Algarve, Campus de Gambelas,
8005-139 Faro, Portugal
pventura@ualg.pt
[2] INESC-ID/Instituto Superior Técnico, Rua Alves Redol,
nº 9, 1000-029 Lisboa, Portugal
alberto.silva@acm.org

Abstract. Within the constant evolution observed in IT/IS area, new processes emerged faced to new customer's requirements and also due to new trends in software engineering community, such as unified or agile processes. Despite the great set of available tools in process and project management, still there is a real gap between "process" and "project" management approaches and respective tools. To assist team members on their work, the effort spend in a process customization could be used in other tasks, such as the project management task to control activities, work products and team members. In this paper, we describe a simplified SPEM-based metamodel for process specification and explain the motivation around this proposal. Considering this metamodel, we also propose a metamodel for project definition and configuration. To conclude, we demonstrate that this metamodel is better adapted to processes specification and can be applied in a project definition.

1 Introduction

The software engineering can be viewed according two basic dimensions: Process Management and Project Management. *Processes management* are pertinent techniques and tools applied to a process to implement and improve process effectiveness, hold the gains and ensure process integrity in fulfilling customer requirements [1].

Project Management is the application of knowledge, skills, tools and techniques to project activities to meet specific project requirements [2]. Project management is accomplished through the application and integration of project management tasks of initiating, planning, executing, monitoring and controlling and closing. The project goals must be followed in terms of costs, time and quality [3].

Process is an activities sequence with a set of inputs and outputs performed by specific roles during a time period. *Project* is a process instance that is performed according process guidelines. The adopted process in a project includes a set of predefined activities that can be planned and monitored in the scope of its own project management. Thus, it is evident an interconnection between process management and project management. The effort applied in a process configuration could be used in other tasks, such as project management tasks in order to control activities, products and team members. The use of efficient processes can help in the success of project execution. However, it is necessary to define process characteristics before transiting

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 516 – 525, 2005.
© Springer-Verlag Berlin Heidelberg 2005

for a project specification. In this paper, we propose a simplified SPEM-based process metamodel. Comparing our model with SPEM, we show that for an interaction between process and project specification, it is better to have a simple metamodel like that we propose.

This paper is organized in the following sections. Section 2 defines the process and project concepts. Section 3 describes related work, which somehow influence our work. Section 4 presents an overview of the ProjectIT initiative, whose main goal is to contribute with new ideas to improve the software development process. Section 5 describes the ProjectIT process metamodel, and presents its architecture and main functionalities, also presents the metamodel for projects definition. Finally, Section 6 summarizes general considerations on present possibilities and limits of the approach.

2 Process and Project Concepts

Process is a set of interrelated work activities characterized by a set of specific inputs and value added tasks that make up a procedure for a set of specific outputs. *Activities* are the units of work that sometimes may be related to each other, e.g., forming a hierarchy of activities, and they are associated to roles. *Roles* describe in an abstract form the set of skills and/or responsibilities associated with the execution of one or more activities. During activity enactment, developers create and transform work products. *WorkProduct* represent the object of work in an environment, and correspond to typical software development objects, such as requirements documents, test plans, test cases, etc. *Tools* automate execution of certain activities [4].

Processes are meant to be instantiated, resulting in an executable entity called a *project* (or simply process instance). The involving task of project coordination is called project management. Project management knowledge and practices are best described in terms of their component processes. These processes can be placed into five process groups (initiating, planning, executing, controlling and closing) and nine knowledge areas (project integration management, project scope management, project time management, project cost management, project quality management, project human resource management, project communications management, project risk management and project procurement management). *Project enactment* is guided by a project plan that might initially correspond to a parameterized instance of a generic process plan that is not necessarily complete. As work progresses, project plans may be adapted to project specific needs, or might be complemented with additional project specific definitions.

A *metamodel* defines a language for describing a specific domain of interest. For example UML (Unified Modelling Language) [5] is a language (i.e., a metamodel) to describe models for engineering systems. Some other metamodels address domains like process, organization, quality of service, etc. A process metamodel provides a set of generic concepts to describe any process models [6].

3 SPEM Metamodel

SPEM metamodel is used to describe a concrete process or a family of related processes [7]. The actual processes enactment, that is, project planning and executing,

is not the scope of the SPEM metamodel. SPEM specification is presented as a UML profile and also provides a MOF based metamodel. This approach results from the large number of process models and standards, using each one a different terminology. SPEM allows different processes specifications, namely XP [8] [9], RUP [10] [11], MSF [12], DMR [7] [13], CMM [14] or ISO [15].

Although, UML is not necessarily tied to any application area or modelling process, its greatest applicability is in the area of object-oriented software design. UML is defined by a metamodel, which is itself defined as an instance of the MOF metametamodel. SPEM metamodel is described in a similar form as an UML subgroup extension called *SPEM_Foundation*. The SPEM metamodel is divided into four packages: *BasicElements*, *Depedencies*, *ProcessStructure*, *ProcessComponents* and *ProcessLifecycle*.

SPEM is a metamodel for process specification, so it is necessary to define its basic concepts in *ProcessStructure*, *ProcessComponents* and *ProcessLifecycle* packages. SPEM principles starts from the idea that a software development process is collaboration between abstract active entities called *Process Roles* that perform operations called *Activities* on concrete, tangible entities called *WorkProducts*.

3.1 SPEM Support Concepts

Fig. 1 defines the main structure elements (package *ProcessStructure*) that serve to construct a process description. In the diagram, the class *WorkProduct* is something that is produced, consumed or modified by a process, can be a document, a model or a source code. *WorkProductKind* describes a product category, could be a text document, a UML model, an executable, a code library or others. For that, each *WorkProduct* has associated a *WorkProductKind*. A *ProcessRole* is associated to each *WorkProduct*, which is formal responsible for its production. *WorkDefinition* is element of a process model that describes execution, operations and transformations performed in *WorkProducts* by the *ProcessRoles*. Its main subclass is *activity*, but *phase*, *iteration* and *lifecycle* are also its subclasses (Fig. 3). *ProcessPerformer* defines the responsible for a set of higher level *WorkDefinitions* in a process where individual roles cannot be associated. Its subclass *ProcessRole* defines responsibilities and abilities on specific *WorkProducts* and indicates the roles that perform or assist specific activities. Class *Activity* represents the work performed by a role, corresponding to tasks, operations and actions assisted or coordinated by a role. The atomic elements that constitute an *Activity* are called *Steps*.

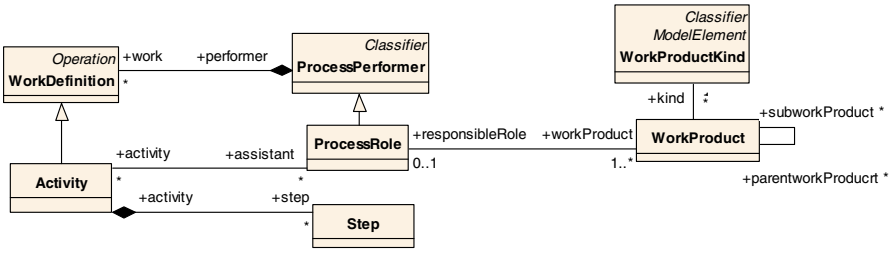


Fig. 1. Process structure (adapted from [7])

3.2 SPEM Process Components

Fig. 2 describes a process components package (sub-package *ProcessComponents*). Its classes are responsible for dividing one or more processes descriptions in "self-contained" parts that can be placed under configuration management or versions control. Such as in UML [5], a *Package* can both own and import process definition elements. A *ProcessComponent* is a process description that is internally consistent and can be reused with other process components to assemble a complete process. *Process* is a ProcessComponent intended to stand as a complete process from which it is possible to specify projects. It's distinguished from normal process components by the fact that it isn't intended to be composed with other components. *Discipline* is a particular specialization of *Package* that partitions the process activities according to a common "theme".

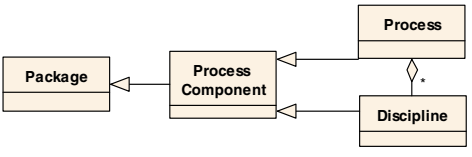


Fig. 2. Process components (adapted from [7])

3.3 SPEM Process Lifecycle

Package *LifeCycle* introduces process definition elements that help to represent its execution over time (Fig. 3). These elements describe or constrain the overall behaviour of the performing process and they are used to assist with planning, executing and monitoring the process. To coordinate its execution, activities order can be restricted. It is necessary to define the "shape" of the process over time and its lifecycle structure in terms of *Phases* and *Iterations*.

Phase is defined with the additional constraint of sequentially; that is, their enactments are executed with a series of milestone dates spread over time and often assume minimal (or no) overlap of their activities in time. A *Lifecycle* is associated with a sequence of *Phases*. An *Iteration* is a composed *WorkDefinition* but with a minor milestone (goal).

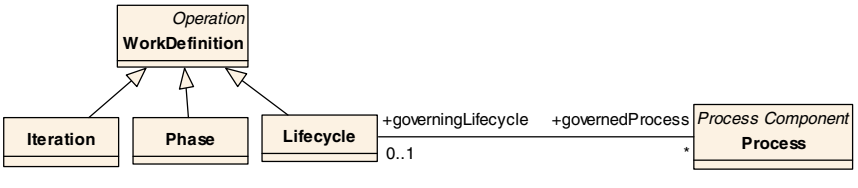


Fig. 3. Process lifecycle (adapted from [7])

4 ProjectIT Initiative

The INESC-ID Information Systems Group is a research group interested in topics related with software engineering and the software development process, and in applying them to the daily projects in which it is involved. ProjectIT is an R&D program that integrates some concrete issues related with information systems design, development and operation problems or, in general, the problematic of "projects in the area of information technologies"[16]. Its main goal is to provide a complete software development workbench, with support for project management, requirements engineering, analysis, design and code generation features.

ProjectIT intends to produce some results, namely (1) a collaborative tool with Web interface (i.e., with Web-client access) called "ProjectIT-Enterprise"; and (2) a rich-client tool (windows based) for improved productivity called "ProjectIT-Studio". Both tools present tight complementarities and integration mechanisms. The Studio version has as its main goal to provide mechanisms for higher productivity to requirements management and specification, models design, automatic code generation and software development. On the other hand, the "ProjectIT-Enterprise" version provides mechanism to collaborative support for team work, emphasizing project management activities, workflows and documents management.

5 ProjectIT Metamodels

Considering the SPEM metamodel and comments like "*The actual enactment of processes—that is, planning and executing a project using a process described with SPEM, is not in the scope of this model*" and "*…minimal set of process modeling elements necessary to describe any software development process, without adding specific models or constraints for any specific area or discipline, such as project management or analysis*" in an OMG document [7], make evidence specific and different needs in process and project models. SPEM address some concepts that are not essential in *project management* scope, these elements are: *WorkDefinition*, *Step*, *ProcessPerformer*, *Lifecycle* and *Guidance*. In project coordination there's no need to represent *WorkDefinition* as a composite pieces of work that are further decomposed in *Activities* and *Steps*. Class *ProcessPerformer* is not necessary considering his connection to *WorkDefinition*. *Lifecycle* is an abstract concept that describes how project time is organized in *Phases* and *Iterations*, without relevance in a project domain. SPEM package *BasicElements* has elements which contain a description of Model Elements, an essential feature in software process but without importance in project management discipline. These observations take us to a new approach considering new metamodels for project management based in SPEM.

PIT-ProcessM (ProjectIT Process Metamodel) is the component that supports, among other features, the definition of process models. On the other side, PIT-ProjectM (ProjectIT Project Metamodel) is a metamodel for projects definition. The main goal is to capture process concepts and mechanisms and apply them in concrete projects such as to monitories activities, evaluate products quality or people productivity. In this paper we present the PIT-ProcessM metamodel that allows the definition of process models and PIT-ProjectM metamodel for project creation.

5.1 Process Metamodel

Fig. 4 presents the PIT-ProcessM architecture, where is evident the relationships between the main process elements. The PIT-ProcessM defines the classes that correspond to elementary process concepts, allowing process creation or modification. Two complementary views show those static and dynamic process elements.

The static view describes *Activities*, *Roles*, *Workproducts* and *Disciplines*. Dynamic view describes how "things" append over time. An interface between this views is made by the class *Activity_Iteration*, where are specified the *Activities* that belongs to an *Iteration*.

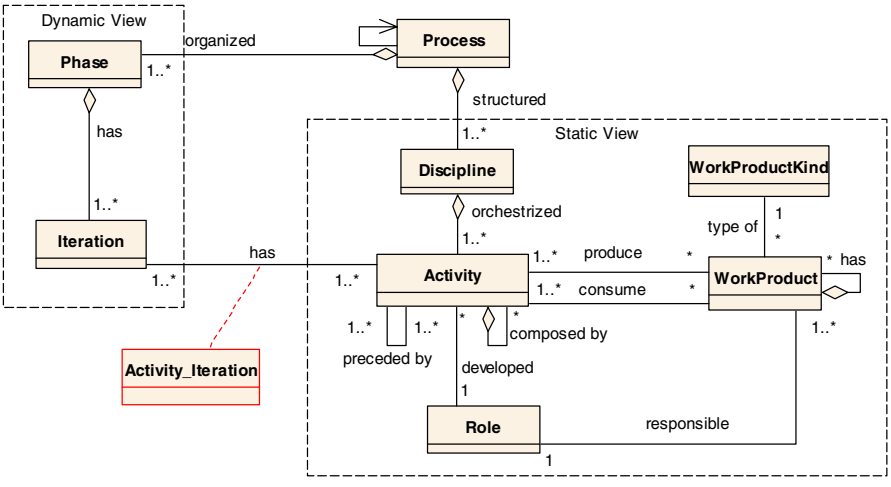


Fig. 4. ProjectIT Process Metamodel (PIT-ProcessM)

Static View. An Activity represents the work performed by a role. But an activity can be decomposed in small work units, also called activities to unlimited deep of nested work. This concept is represented by the reflexive composed by aggregation. Control and data flow between activities is defined by the reflexive preceded by association. Activities produce and consume WorkProducts, which can also be formed by a set of small WorkProducts. Each WorkProduct is identified by a WorkProductKind, e. g., a document, a model, a source code, and so on. Activities are organized according to a common "theme" in Disciplines.

Dynamic View. The dynamic view identifies how process can be managed in terms of phases and iterations. Phases are defined with the additional constraint of sequentiality, with a series of milestones spread over time and often assume minimal overlap of their activities in time. Each phase include some iterations, which are activities flows but with small goals.

5.2 Project Metamodel

A project specification is based on the initial process definition model. *Projects* are frequently divided into more manageable components or subprojects, although the

individual subprojects can be referred by themselves as projects and managed as such. *Projects* have associated specific information on *phases*, *iterations*, *activities* and *workproducts*. In a modelling perspective (Fig. 5) the differences between PIT-ProjectM and PIT-ProcessM metamodels are in the following classes: (1) *ActivityIterationProject,* which defines the activities to perform in each iteration; (2) *Person*, team members; (3) *PersonRoleProject* that defines the association between persons and project roles.

Associations between the classes *Person*, *WorkProductProject* and A*ctivityIterationProject* allow each person to visualize its responsibilities (activities and products), and verify relations to other persons work.

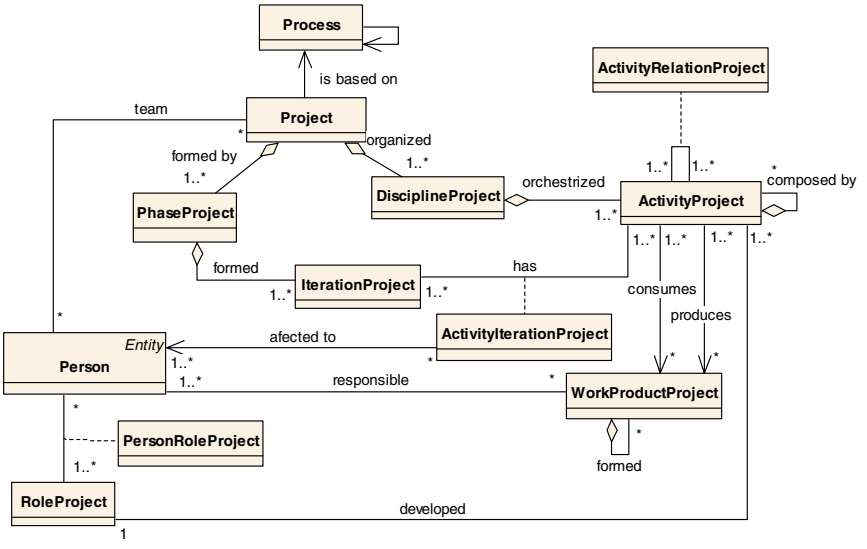


Fig. 5. ProjectIT Project Model (PIT-ProjectM)

Team members can manage their time, place priorities in their activities, made decisions supported by data. Project manager has a global vision of project performance, being able to observe details in activities, iterations and team performance.

6 Application Example with a XP Process

The following provides an application example of the proposed metamodel. For brevity, the example only covers the PIT-ProjectM metamodel for an agile project based on XP process. As seen in Fig. 5, the project model is based on the process defined according to PIT-ProcessM metamodel (Fig. 4).

The metamodel must be applied in all project Releases (*PhaseProject*) and Iterations. As an example we just considered the *fist Iteration* of *Release 1* and only one discipline (Planning). For this reason, classes *PhaseProject*, *IterationProject* and *DisciplineProject* are not instantiated in the UML diagram of Fig.6. The association

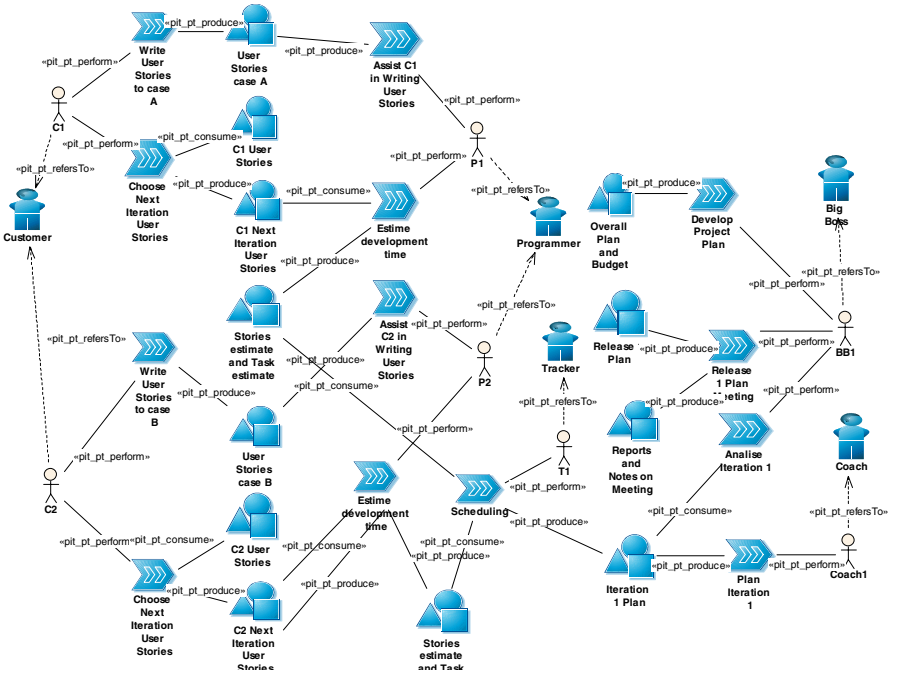


Fig. 6. Example of a XP project diagram based on PIT-ProjectM metamodel

that defines the *Person* responsible for a *WorkProductProject* is also omitted. Dependencies between project activities are not showed in this kind of diagram. Another UML activity diagram must be considered to represent relations between activities represented by class *ActivityRelationProject*.

The UML diagram of Fig. 6 represent *ActivityProject* performed by *Persons* (team members) and *WorkProductProject* produced or consumed by *ActivitiyProject*. Each *Person* only perform the kind of activities (*ActivityProject*) assigned to his *Role*. To understand the meaning of class symbols, a situation is described. The *Person* C1 (whose *Role* is a Customer) performs "Write Users Stories case A" and "Choose Next iteration User Stories" (*ActivityProject*). In the first activity the *WorkProductProject* produced are "User Stories from case A". In the second activity, C1 takes his User Stories and selects the User Stories to be programmed in the next Iteration.

7 Conclusions and Future Work

Table 1 describes relations between concepts of SPEM and PIT-ProcessM metamodels.

SPEM metamodel is the standard for process specification, without specific models or constraints for any specific area or discipline, such as project management or analysis. Some SPEM basic elements are not important in a project management perspective. Elements like *WorkDefinition*, *Step*, *ProcessPerformer*, *Lifecycle* and

Table 1. Common concepts to SPEM and PIT-ProcessM metamodel

	Dynamic View			Static View			
SPEM	Phase	Iteration	Discipline	Activity	Step	Process Role	WorkProduct
PIT	Phase	Iteration	Discipline	Activity	Activity	Role	WorkProduct

Guidance are not relevant in planning, scheduling, collaboration and communication. The metamodel structure organized in four packages make clear that an interconnection between their elements to support a project definition would result in a complex model. As an example, SPEM metamodels doesn't address a direct relationship between *disciplines* and *activities*. But in a project management perspective, it is essential to organize *activities* to make an easier project tracking.

If the metamodel is simplified, extension mechanisms could be adapted to specify project management issues integrated with metamodel elements. An important organizational feature to project management is team skills management, so it's essential to extend the models but keep them simple. Ours simplified SPEM based metamodels - PIT-ProcessM and PIT_ProjectM - can be applied in a process and project integrated environment to manage projects and teams. These UML models can be applied in a collaborative tool to support project management.

In this paper, we discussed some process and project concepts and their relationship to SPEM and PIT metamodels. We also demonstrate that a simplified process metamodel is better to define project features essential in project management.

References

1. Roger S. Pressman, "Software Engineering – A Practitioner´s Approach", McGraw Hill, 5th Edition, November 2003
2. David I. Cleland, William R. King, "Project Management Handbook", New York: Van Nostrand Reinhold, 1988.
3. Sommerville, "Software Engineering", Addison Wesley, 7th Edition, May 2004
4. P. Barthelmess,"Colaboration and coordination in process-centered software development environments: a review of the literature", I*nformation and Software Technology*, Elsevier, 2003, pp. 911-928.
5. OMG, "OMG Unified Modelling Language Specification version 1.3", OMG Document formal/02-11-14, November 2002.
6. C. Rolland, S. Nurcan, G. Grosz, "Enterprise knowledge development: the process view", Information & Management, vol. 36, pp. 165-184, 1999.
7. OMG, "Software Process Engineering Metamodel Specification", Version 1.1, January 2005, http://www.omg.org/technology/documents/formal/spem.htm (accessed in May of 2005).
8. K. Beck, "Extreme Programming Explained". Boston, MA: Addison-Wesley, 2000
9. R. Jeffries, A. Anderson, C. Hendrickson, "Extreme Programming Installed", Addison Wesley, 1st Edition, October 2000.
10. P. Kruchten, "The Rational Unified Process: An Introduction", Addison-Wesley Pub Co, 3rd Edition, December 2003.

11. Rational Unified Process (RUP), Rational Unified Process, Version 2003.06.01.
12. G. Lory, "Microsoft Solutions Framework", Version 3.0, http://www.microsoft.com/technet/itsolutions/techguide/msf/msfovrvw.mspx (accessed in June of 2004).
13. DMR Consulting, "DMR Macroscope", Version 3.1, April 2000.
14. CMM, Capability Maturity Model for Software, http://www.sei.cmu.edu/cmm (accessed in June of 2004).
15. ISO, ISO/ICE 12207, http://www.software.org/quagmire/descriptions/isoiec12207 (accessed in June of 2004).
16. Alberto Rodrigues da Silva, "O programa de Investigação Project-IT", Technical report, V1.0, October 2004, INESC-ID, http://berlin.inesc.pt/alb/uploads/1/193/pit-white-paper-v1.0.pdf.

Requirements for Secure Logging of Decentralized Cross-Organizational Workflow Executions

Andreas Wombacher[1], Roel Wieringa[1], Wim Jonker[1],
Predrag Knežević[2], and Stanislav Pokraev[3]

[1] University of Twente, 7500 AE Enschede, The Netherlands
{A.Wombacher, R.J.Wieringa, Jonker}@utwente.nl
[2] Fraunhofer Gesellschaft,
Integrated Publication and Information Systems Institute,
64293 Darmstadt, Germany
knezevic@ipsi.fhg.de
[3] Telematica Institute, Enschede
Stanislav.Pokraev@telin.nl

Abstract. The control of actions performed by parties involved in a decentralized cross-organizational workflow is done by several independent workflow engines. Due to the lack of a centralized coordination control, an auditing is required which supports a reliable and secure detection of malicious actions performed by these parties. In this paper we identify several issues which have to be resolved for such a secure logging system. Further, security requirements for a decentralized data store are investigated and evaluated with regard to decentralized data stores.

1 Introduction

A multi-lateral collaboration representing a cross-organizational workflow, is based on communication between several parties each providing its own local workflow. However, several interaction structures of such workflows can be differentiated [1]. The most different once are centralized and decentralized workflows. A centralized workflow consists of a single coordinator, who derives and checks the actions to be performed next. In a decentralized cross-organizational workflow such a centralized coordinator is missing and all parties have to trust the other parties to perform correct actions, that is, actions which are in accordance with the cross-organizational workflow. If an action is performed[1] which (i) does not fit to the receiving party's workflow state or (ii) an action derived from a party's workflow state is not accepted by the receiving party, then an inconsistency of the cross-organizational workflow state is detected in case the cross-organizational workflow is considered to be consistent[2].

However, there are two main strategies to cope with inconsistent states in cross-organizational workflows: the first option is to prevent these states to happen and the second option is to provide means to detect these inconsistent states and to determine the fraudulent party (detection approach). Prevention has been addressed in different

[1] An action which is not performed results in later actions, which do not fit to the cross-organizational workflow and therefore must not be considered explicitly.
[2] Consistency here means the deadlock-freeness of the cross-organizational workflow.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 526–536, 2005.
© Springer-Verlag Berlin Heidelberg 2005

approaches e.g. under the label of fair exchange of goods in multi-lateral collaborations [2,3]. However, these approaches require a lot of shared knowledge of the cross-organizational workflow and its state to provide the requested properties. Alternatively, non-repudiation protocols can be used to resolve conflicts in a bilateral exchange, although this does not provide a solution for the multi-lateral case due to a lack of a global ordering criteria of the bilateral exchanges [3]. Further, all these approaches introduce a lot of communication overhead, which in most of the cases is not needed due to the partners act honestly. Therefore we follow the detection approach supporting a logging information to determine actors and malicious actions, which are unsupported by the cross-organizational workflow. This decision has to be derived based on the logged local state information of the parties involved and being relevant to this instance of the cross-organizational workflow execution. The overall aim is to provide a logging mechanism where local state information is used to reconstruct the cross-organizational workflow state and to determine the party performing a malicious action. As a consequence of this, integrity and non-repudiation of logging data has to be ensured. Further, since logging data provides a detailed insight into the mission critical information of a party data privacy has to be granted.

The aim of this paper is to identify the issues and security requirements of auditing cross-organizational workflows and discuss potential logging methods and their compliance with the identified security requirements. In particular, we illustrate that the cross-organizational workflow state construction problem is independent of the used storage method. Further, we elaborate that centralized logging is hard to realize in concrete applications and that Distributed Hash Tables with probabilistic data availability guarantees cover the specified security requirements.

The paper starts with a brief introduction of a scenario (Section 2) used for explaining the state representation and communication issues (Section 3). In Section 4 the security requirements for a decentralized data store are derived. In Section 5 the derived security requirements are compared with decentralized storage solutions. Finally the paper concludes with related work, a summary and future work.

2 Scenario

Auditing cross-organizational workflows is discussed by means of an order process example consisting of an accounting department authorizing the order after authenticating the buyer submitting the order and a logistics department providing the shipping including a parcel tracking.

The cross-organizational workflow representing this scenario as depicted in Figure 1 is represented in Finite State Automaton (FSA)[5] notation [4]. States are represented as circles, transitions represent message exchanges denoted as arcs where sender, recipient and message name are annotated to an arc. The successful termination state of an FSA is indicated by a final state denoted by a circle with a thick line.

[3] A classical result of distributed systems is that bilateral knowledge is insufficient for deriving global knowledge due to loss of information. Global knowledge can only be derived if additional information is provided. [4]

[4] More complex models like Workflow Nets (WF-Net) [1] could also have been used.

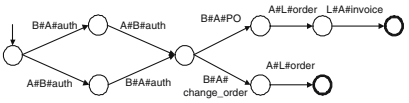


Fig. 1. Cross-organizational Workflow

The process starts with buyer and accounting department authenticating each other, while both orders of authentication are supported using *B#A#auth* and *A#B#auth* messages. After the authentication the buyer sends either the purchase order (*B#A#PO* message) or a change order request (*B#A#change_order* message) to the accounting. The buyer request is the basis for the accounting to send an order (*A#L#order* message) to the logistics department. In case the order is a new order, the logistics requests money from the accounting by sending an invoice (*L#A#invoice* message).

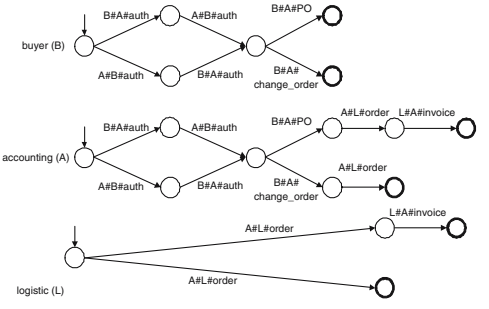


Fig. 2. Public Workflow Models

The cross-organizational workflow depicted in Figure 1 represents the combined public workflows of the three parties involved. In particular, the public workflow describes the externally observable message exchanges the party is involved in during the execution of the cross-organizational workflow. The public workflows of the three parties are depicted in Figure 2.

In this example only three parties are involved in an instance of the cross-organizational workflow. However, it is usual that the different parties are participating in several instances of a workflow and are involved in several cross-organizational workflows, thus, there is a high number of parties forming a network. A characteristic of this network is that it is changing over time, that is, new parties are joining, while others are leaving. To provide a logging of the cross-organizational workflow a decentralized representation of the cross-organizational workflow state and a decentralized data store are required.

3 State Representation

The aim of the state representation is to be able to derive the state of the cross-organizational workflow based on the locally logged state information. Since the cross-organizational

workflow is coordinated by the exchange of messages, the only state information which can be interpreted by all parties is the sequence of exchanged messages. The challenge is to reconstruct the message sequence of the cross-organizational workflow from the logged public workflow message sequence information.

3.1 Communication Model

The way to derive cross-organizational state information from public one dependents on the communication type: synchronous communication, where the recipient must pick up the message before the sender can continue the processing, or asynchronous communication, where the sender does not need to wait for reception of the message before continuing. Due to this definition, the order of sending messages and the one of receiving messages may quite vary in asynchronous communication model, while they are quite the same in the synchronous communication model. Since the synchronous communication model contains more constraints it is easier to handle and will be used for further discussion. In particular, message sequences supporting a synchronous communication model can be represented as Finite State Automata (FSA) [5].

3.2 Example

Let's assume first there are no malicious actions contained in the public workflow's message sequences. Then the construction of the cross-organizational workflow's message sequences is based on combining the public workflow's message sequences by keeping the local order of the public sequences and using the co-occurrence of sending and receiving messages to further reduce the potential number of combinations. Let's consider the following execution sequences derived from the public workflows depicted in Figure 2, where a message is represented by its sender S, its recipient R, and the message name msg as $S\#R\#msg$:

$$\begin{array}{ll} B: & A\#B\#auth- \ B\#A\#auth- \ B\#A\#PO \\ A: & A\#B\#auth- \ B\#A\#auth- \ B\#A\#PO- \ A\#L\#order \\ L: & A\#L\#order \end{array}$$

combing the message sequences supported by B and L results in the following potential sequences:

$$\begin{array}{llll} A\#L\#order- & A\#B\#auth- & B\#A\#auth- & B\#A\#PO \\ A\#B\#auth- & A\#L\#order- & B\#A\#auth- & B\#A\#PO \\ A\#B\#auth- & B\#A\#auth- & A\#L\#order- & B\#A\#PO \\ A\#B\#auth- & B\#A\#auth- & B\#A\#PO- & A\#L\#order \end{array}$$

considering also the sequence reported by A and combining it with the above derived potential sequences constructed from the sequences supported by B and L results in the single sequence

$$A\#B\#auth - B\#A\#auth - B\#A\#PO - A\#L\#order$$

which represents the single possible state of the cross-organizational workflow. The combination of the public message sequences not necessarily results in a single sequence,

but may result in a set of sequences in case the process is not completed and further actions are still possible. However, a complete execution of a cross-organizational workflow always results in a single combined execution sequence.

3.3 Concurrent Communication

There are rare cases where the order of sending and receiving messages do not correspond as illustrated on behalf of the following example. The buyer and the accounting department can concurrently exchange the authentication messages, thus, the messages may cross each other at the communication channel. The following execution sequences can be derived from the public workflows, which individually conform to the cross-organizational workflow.

$$\begin{array}{ll} A: & A\#B\#auth- \; B\#A\#auth \\ B: & B\#A\#auth- \; A\#B\#auth \end{array}$$

With regard to the cross-organizational workflow, these two sequences can not be combined to a single one as described in Section 3.2, because the order of the messages are contradicting each other and a combined order of the two partial orders can not be achieved. As a consequence, two potential message sequences have to be considered, that is, $A\#B\#auth - B\#A\#auth$ and $B\#A\#auth - A\#B\#auth$ where a precedence of the ordering of one party is given against the ordering maintained by the other party. To avoid the handling of these precedences the following issue has to be resolved:

Issue 1. *(Exclusive Communication Channels) The used synchronous communication model must guarantee that the bilateral communication channel is used exclusive, that is, can only be used by a single party at a time.*

In case this issue is not resolved, the resulting state representation problem is comparable to the one observed under the asynchronous communication model.

3.4 Malicious Actions

In the following we consider parties to log malicious actions under the prerequisite that the cross-organizational workflow is consistent, that is, does not contain deadlocks. The following cases involving malicious actions are observable for a single message exchange:

A single party logs malicious actions, while the other one logs truthfully. It can be detected that two different messages have been logged, although it can not differentiated which party cheated. It could either be the sender, who logged a message he hasn't sent to the recipient, who is logging truthfully, or the sender logging truthfully while the recipient logs a malicious action. It get's even worse if both parties cheat.

Both parties log malicious actions. In case the parties are logging different messages, again the difference can be detected although it can not be decided who or whether at least one acted truthfully. In case the two parties agree on which malicious action to log (the two parties conspire), then the malicious logging remains undetected in a first step. Since each party can log any message, the representation of a message exchange in

the cross-organizational workflow depends on the information of the sent and received messages. The Hamming distance h specifies the difference between two different codings or code words, where the distance provides information about the error detection (maximum $h-1$ errors can be detected) and error correction (maximum $\frac{h-1}{2}$ errors can be corrected) capabilities [6]. With regard to the reconstruction of the global state, two logging entries form up an entry for the global state, therefore the Hamming Distance is two supporting the detection of one error and no error correction.

We observe this behavior in the first case involving malicious actions, where only one party cheats. Here we can detected the error, but can not decide who caused the error to correct the cross-organizational state information. As a consequence, the limitation to one error detection and no error correction is quite strong and is not appropriate for the problem of auditing cross-organizational workflows. The research challenge here is how to increase the Hamming distance of the log data or how to use additional context information to support error correction and higher order error detection.

Issue 2. *(State Representation) The data logged by the different parties has to be configurable with regard to the number of resolvable malicious actions in accordance to the actual security requirements of the application scenario.*

3.5 Conflict Resolution

Before further digging into the technical issues of the logging and the corresponding security requirements for storing log data, we will discuss the approach of conflict resolution using the log data. The process starts with a party complaining about state inconsistency observed at its local state representation, that is, receiving a message which does not fit to the current state or a party refusing to accept a sent message. In either case, an authority is included, which is trusted by all parties like e.g. court. In front of the authority, each party presents his evidence on illustrating that he/she always acted truthfully. The authority checks the proof for integrity and combines the proofs to resolve the conflict, that is, identifies the party performing malicious actions. Finally, the authority specifies the compensation to be provided by the misbehaving parties to the remaining parties.

One major issue here is to provide reliable proofs derived from the logging information. A standard approach to maintain logging information is using a trusted third party, that is, a single party which is trusted by all parties involved in the cross-organizational workflow. An example of such an approach is e.g. [7] or [8], but there also the coordination of the cross-organizational workflow (at least the general one) is performed by a centralized instance as opposed to a decentralized coordination as discussed in this paper. Further, the centralized solution suffers from the fact that all involved parties have to agree on a single trusted third party. In particular, [9] states that decentralization engenders trust and centralized regulation destroys it, which favors a decentralized monitoring of state information. Therefore, we focus in the following on decentralized storage models for logging information and the discussion of the security requirements. However, the above discussed issues of exclusive communication channels (see Issue 1) and state representation (see Issue 2) are independent of the used storage model for logging data.

4 Security Properties of Data

Independent of the format and the details about the actual data being logged there are several security requirements which have to be considered using a decentralized auditing of cross-organizational workflows: integrity of public workflows and data, privacy of content, originator or storing party, and data availability [5].

4.1 Integrity of Public Workflow Models

The basis for comparing the log information with the actual cross-organizational workflow are the public workflows being the basis for the cross-organizational one. It is important that the public workflows are not altered after the consistency of the cross-organizational workflow has been confirmed. Thus, the following issue arises:

Issue 3. *(Integrity of Public Workflows) The integrity of the public workflows used for consistency checking has to be guaranteed.*

A potential approach to this is to log the public workflows using the same data store as for logging message exchanges. For example, each party logs its own public workflow and the ones of its trading partners using the same mechanisms as for logging data.

4.2 Integrity of Stored Data

The usage of a decentralized data store for logging data implies that log information is handled by several parties until it is finally stored. All parties involved in the storing or retrieval procedure may tamper the integrity of the data passed by. In particular, the data can be modified, data can not be forwarded in an appropriate way, that is, withdrawing data from the storage system, or false data can be introduced by replacing the original data, as e.g. done in replay attacks. The latter two address transactional properties of the store and retrieve operations of the used data store.

The idea is to have no need to secure the store and retrieval operations, because the originator and the storing party remain unknown/private. Therefore, no party in the middle performing a part of the operation knows whose data it is currently handling therefore does not provide any information to perform specific threads. Obviously, a party may block or change data passing by, but this is detectable by the originator of the operation. Such a party can quite easily be identified due to determinism of the used operations. Therefore the integrity requirements on data drill down to integrity of data content:

Issue 4. *(Integrity of Data) The decentralized data store has to support the integrity of data content and transactional properties.*

[5] In the following we assume reliable communication channels, since unreliable communication channels introduces several orthogonal problems, which are addressed e.g. by the distributed systems community (see e.g. [4]).

4.3 Privacy

Due to the usage of a decentralized data store resulting in storing content at different parties, which are unknown to the party performing the store operation, the **privacy of the data content** has to be protected. In particular, it has to be ensured that the content could neither be read by the party routing the content nor by the party storing the content. Further, **privacy about the originator** of the store operation as well as **privacy about the actual storing party** is required. The first one prevents that a storing party alters the content or doesn't log the content to foist a malicious action on the originating party initiated by the storing party itself. The second one prevents the party performing the store operation to conspire with the storing party to finally alter data, add or remove data. Thus, the privacy requirements can be summarized as follows:

Issue 5. *(Privacy Requirements) The decentralized data store has to provide privacy of data during routing and storing.*

4.4 Availability of Data

Since the decentralized data store relies on the parties storing locally the logging information, the disappearance of a party results in unavailable data. Although, the loss of data either temporarily or permanent is not acceptable in our scenario. However, a guarantee of data availability can not be provided due to the nature of decentralized systems, but the remaining risk can be estimated and the data store can be adapted in accordance. Thus, we require probabilistic guarantees on the data availability.

Due to long running transactions there is no option to specify when a log information will never be required afterwards, therefore an explicit deletion of data is required, although it has to be prevented that logs can be deleted before the completion of the process by all involved parties. Thus, everybody has to agree that the business process has been completed and that the corresponding logging information of that business case can be deleted. In particular, this requirement covers two orthogonal requirements: the first one addresses a consensus making of the involved parties on deleting the log data, while the second one is performing the delete operation in a decentralized data store. We summarize these observations as the requirements on the data store:

Issue 6. *(Availability) The decentralized data store has to support availability of data in fluctuating networks with probabilistic guarantees.*

and an issue on the organization of the cross-organizational workflow initialization and termination phase:

Issue 7. *(Consensus on Deletion) The logging system has to support deletion of log data only after a consensus of all parties involved in the collaboration to delete the data has been reached.*

5 Monitoring Solutions

Based on the above investigation of security requirements on the decentralized data store, a brief overview of potential solutions is provided.

5.1 Decentralized File Sharing

Current decentralized storage systems can be roughly divided into the following categories: file-sharing and decentralized storage approaches. The first file-sharing approaches have been Napster followed by Gnutella, KaZaA and eDonkey [10], where every peer makes its files available to the community. They are content-sharing applications, a file-sharing subgroup. Another approach to file-sharing are content-storing approaches, which enable file access for the given community. In particular, every peer offers its disk space that is used by other peers for storing their files. Examples of that category are FreeNet, GNUNet, Past or OceanStore [10].

As a consequence of a thorough investigation of decentralized file sharing, which can not presented here due to a lack of space, it can be derived that content-sharing approaches are inappropriate since they do not provide the required integrity of the logged data. Although, the content-storing approaches provide this capability as well as some support for high data availability exists partially, the privacy requirement of the storing party is not provided.

5.2 Distributed Hash Table

Further research in decentralized systems proposed Distributed Hash Tables (DHT) as the next generation low-level decentralized storage approach. DHTs are quite mature and many implementations like Pastry, Tapestry, P-Grid, CAN and Chord [10] are available.

Distributed Hash Tables (DHT) are low-level structured decentralized systems that provide a consistent way of routing information to the final destination, can handle the changes in topologies and, have an API similar to the hash table data structure. Thus, log data (representing the content object) can be stored in a DHT by using a key (e.g. generated by a hash function).

Due to the peer-to-peer communication the originator of the content and the final storing peer remain unknown to the peers in general. There are some options to derive partial knowledge of especially the storing peer derived from the routing tables maintained locally, but this option is limited to a fraction of the key space only. Thus, the privacy requirement (see Issue 5) seems to be covered in most of the cases by this approach.

Unfortunately, a DHT layer as available right now does not guarantee the availability of data it manages. Whenever a peer goes offline, locally stored data become inaccessible. However, there are approaches recently under development, where higher data availability can be reached by enabling redundancy in two ways: replicating data many times, or using erasure coding to code data and dividing them into many blocks. Although erasure coding provides in general lower storage costs [11], it has has been argued in [12] that the resulting system design get's more complicated. At a first glance the aim is to start with a more simple approach and applying optimizations after the remaining issues have been resolved. Therefore, we choose replication as a way to achieve data availability in a DHT as required for the auditing (see Issue 6). A potential solution providing probabilistic guarantees has been proposed in [13]. Therefore, the easiest case is to have a DHT with insertions only, but providing an option to delete entries e.g.

by maintaining a revocation list or using a similar approach. As a consequence, the integrity requirement (see Issue 4) can easily be achieved.

6 Related Work

In contrast to the approach for modeling cross-organizational workflows using a sysnchronous communication model, cross-organizational workflow models based on asynchronous communication have been proposed e.g. by v.d.Aalst [15] or Martens [16]. However, these workflow models do not support decentralized decision making due to the complexity of the selected formal model.

Alternative approaches of reconstructing global states based on local information are known from distributed computing, where a group of parties wants to know the system state. The challenge here is to provide sufficient state information to all parties without influencing the overall system too much by the additionally introduced communication. A class of solutions is known as snapshot protocols, like e.g. [17], where state information is reduced to safe points representing the end points of a concrete transaction to be used for synchronization. A similar problem is the multi-party private computation problem [18] addressing the issue of computing a function value without revealing the private parameters. Again, here all parties involved in the calculation can start deriving the function value at any point in time. The difference to the one addressed in this paper is that the cross-organizational workflow state needs only to be reconstructed in case of a complaint as opposed to a continues provisioning of this service.

7 Conclusion and Future Work

The paper identifies issues related to secure logging of decentralized cross-organizational workflows and underlying decentralized data storage. While the latter ones are investigated with regard to content-sharing, content-storing and Distributed Hash Tables, the issues related to the used communication model, the state representation and the number of resolvable malicious actions, the integrity of the public workflows, and the deletion of logging data based on a consensus of all parties have not been addressed in this paper. In particular, these issues will be investigated in future work as a basis for a secure logging system. Please be aware that the issues raised so far are related to static workflows, that is, workflows not changing over time. Obviously, dynamic workflows introduce an additional complexity to the monitoring as discussed in this paper.

References

1. W. v.d. Aalst. Interorganizational workflows: An approach based on message sequence charts and petri nets. *Systems Analysis - Modelling - Simulation*, 34(3):335–367, 1999.
2. N. Asokan, Matthias Schunter, and Michael Waidner. Optimistic protocols for fair exchange. In *Proc. 4th ACM Conf. on Computer and Communications Security*, pages 7–17, 1997.

3. Joachim Biskup and Thomas Leineweber. State-dependent security decisions for distributed object-systems. In *Proc. 15th annual working conf. on Database and application security (DAS)*, pages 105–118, 2001.
4. Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.
5. J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley, 2001.
6. Hans Tzschach and Gerhard Hasslinger. *Codes fuer den stoerungssicheren Datentransfer*. Oldenburg Verlag, 1993.
7. C. Schuler, R. Weber, H. Schuldt, and H.J. Schek. Scalable peer-to-peer process management - the OSIRIS approach. In *Proc. IEEE Int. Conf. on Web Services*, pages 26–34, 2004.
8. Steven P. Ketchpel, Hector Garcia-Molina, and Andreas Paepcke. Shopping models: A flexible architecture for information commerce. In *Proc. 2nd ACM Int. Conf. on Digital Libraries, July 25-28, 1997*, pages 65–74, 1997.
9. Ernest Gellner. Trust, chesion, and the social order. In *Trust: Making and Breaking Cooperative Relations*, pages 142–157. Basil Blackwell, 2000.
10. Dejan Milojičić, Vana Kalogeraki, Rajan Lukose, Kiran Nagaraja, Jim Pruyne, Bruno Richard, Sami Rollins, and Zhichen Xu. Peer-to-peer computing. Technical Report 57, HP Labs, 2002.
11. Hakim Weatherspoon and John Kubiatowicz. Erasure coding vs. replication: A quantitative comparison. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 328–338, 2002.
12. Rodrigo Rodrigues and Barbara Liskov. High availability in DHTs: Erasure coding vs. replication. In *IPTPS '05: International Workshop on Peer-to-Peer Systems*, 2005.
13. Predrag Knežević, Andreas Wombacher, and Thomas Risse. Enabling high data availability in a DHT. In *Proc. of the Intl. Workshop on Grid and Peer-to-Peer Computing Impacts on Large Scale Heterogeneous Distributed Database Systems (GLOBE)*, 2005.
14. Predrag Knežević, Andreas Wombacher, and Thomas Risse. Highly available DHTs: Keeping data consistency after updates? In *Proc. 4th Int. Workshop on Agents and Peer-to-Peer Computing (AP2PC)*, 2005.
15. W.M.P. van der Aalst and M. Weske. The P2P approach to interorganizational workflows. In *Proc. 13. Int. Conf. on Advanced Information Systems Engeneering (CAISE)*, 2001.
16. Ekkart Kindler, Axel Martens, and Wolfgang Reisig. Inter-operability of workflow applications: Local criteria for global soundness. In *Business Process Management, Models, Techniques, and Empirical Studies*, pages 235–253. 2000.
17. K. Mani Chandy and Leslie Lamport. Distributed snapshots: determining global states of distributed systems. *ACM Trans. Comput. Syst.*, 3(1):63–75, 1985.
18. Matt Franklin and Moti Yung. Varieties of secure distributed computing. In *Proc. of the Sequences II, Methods in Com., Security and Computer Science*, pages 392–417, 1996.

Access Control Model for Inter-organizational Grid Virtual Organizations

B. Nasser, R. Laborde, A. Benzekri, F. Barrère, and M. Kamel

Université Paul Sabatier, IRIT/SIERA,
118 Rte de Narbonne F31062 Toulouse Cedex04, France,
Telephone: +33 (0) 5 61 55 60 86, Telecopy: +33 (0) 5 61 52 14 58
{nasser, laborde, benzekri, barrere, mkamel} @irit.fr

Abstract. The grid has emerged as a platform that enables to put in place an inter-organizational shared space known as Virtual Organization. The Virtual Organization (VO) encompasses users and resources supplied by the different partners for achieving the VO's creation goal. Though many works offer solutions to manage a VO, the dynamic, on the fly creation of virtual organizations is still a challenge. Dynamic creation of VOs is associated with the automated generation of access control policy to trace its boundaries, specify the different partners' rights within it and assure its management during its life time. In this paper, we propose an OrBAC (Organization Based Access Control model) based Virtual Organization model which serves as a corner stone in the VO creation automated process. OrBAC framework specifies the users' access permissions/interdiction to the VO resources, where its administration model AdOrBAC flexibly models the multi-stakeholder administration in the Grid.

1 Introduction

In today's demanding business and the rapid technological advancement, organizations tend to compete by investing and excelling in strategic products or services rather training and developing dedicated skills in several domains concurrently. Massive data volumes treatment, long complex computations, unique or critical resources exploitation are examples of applications that necessitate special expertises and resources to deal with. Enterprises are inclined to form supply chains of specialized partners outsourcing business processes via cooperation agreements. Grid technologies played an important role as an infrastructure for realizing such internet-based collaborations [3,4,17]. The grid, from a user's point of view, is a virtual powerful device based on coordinated resources supplied by different partners. However, from administrative point of view, it is a dynamic sharing space called Virtual Organization (VO) that includes users, resources and sharing relationships [1,3,4,5].

Building a VO, in an open environment as the Internet, necessitates enclosing its boundaries. These boundaries encircle legitimate users and resources introduced by the different partners. In addition, it entails sharing relationships which specifies the way and the contexts of usage as: Physicians may *use* Computing-Device and *store* data within *work time*. As a way to achieve that, access control policies may be employed to restrain resources access (computing device, storage space) to legitimate

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 537 – 551, 2005.
© Springer-Verlag Berlin Heidelberg 2005

users (physicians) in certain contexts (work time). However, in addition to the large number of users and resources, the complexity of access control in Virtual Organizations rises from the fact that multiple access stakeholders are involved [5,8]. Each partner needs to keep control on his local assets (add, remove, modify permissions) without compromising discretion or delegating administration to other domains [5]. For example, a client domain needs to manage his users without referring back to the resource provider. To sum up, a VO turns out to be a shared space with multiple access stakeholders, dynamic users and dynamic resources.

Current VO management solutions offer mechanisms to enforce access control policies. These solutions propose the use of access control lists [17] at the different resources sites or employ identity certificates with an out-of-band access control policy [6,7,8]. This vagueness in dealing with policy (static out of band) is unsuitable with the VO dynamics where security policy inconsistencies arise upon dynamically adding or removing users and resources. Policy management becomes essential to prevent security breaches and this in its turn necessitates that partners administrative responsibilities be clear and well specified. We propose a methodology to automate VOs creation starting first with the discovery of potential providers, acquiring access rights to certain resources and finally enforcing the access control policy on runtime upon resources allocation. In this paper, we focus on an access control management model that constitutes the corner stone on the way to dynamic VO creation and management [1]. The selection of an access control model is done according to the identified criteria of selection which are: the need to separate policy from the dynamic infrastructure and the separation of administrative tasks among the different partners.

Based on these criteria, an access control model should be chosen to specify the users-resources relationships. Meanwhile its administration model specifies partners' administrative relationships and thus VO access control policy creation and management. Comparing different access control models as DAC, MAC, RBAC, CBAC(Coalition Based Access Control) [9,10,11,18], we argue that OrBAC (Organization based Access Control) [12] is the most appropriate in our context. ORBAC abstracts users, actions and objects with role, activity and view where its administration model AdOrBAC [14] flexibly models the different administration responsibilities within an organization. This paper starts by introducing the grid environment and its virtual organization notion. In section 3 we cite some of the grid environment characteristics relevant to the access control. To well place our work we show the VO life cycle in section 4 to discuss the choice of an access control model in section 5. Sections 6 introduces OrBAC access control model and AdOrBAC its administration model. In section 7 we show the VO model where in section 8 we discuss some modeling issues and finally we conclude with future works.

2 The Grid

The term "Grid" was adopted in resemblance to the electric grid to designate a powerful system where a pool of distributed devices confederates to allow an authorized user to plug for on-demand energy. However in computer context, energy is supplied in the form of computer hardware and software for problem solving as computing, data storage or applications sharing. The grid has emerged as a platform that has as

goal enabling coordinated resource sharing and problem solving in dynamic multi-domain virtual organization [5].

The different organizations in a grid environment federate into an alliance for a specific goal (ex. project cooperation, resources rental or application provider). They put in place what we call a Virtual Organization (VO) which constitutes the shared space between the different partners. An organization may need to participate in multiple Virtual Organizations according to its needs and thus have multiple shared spaces with multiple communities. Consider the example in Figure 1, a physicist at "Lab1" uses a simulation application at laboratory "Lab2" and then analyzes the data using computation cycles hired from "Enterprise3" and finally stores the whole results at a storage provider.

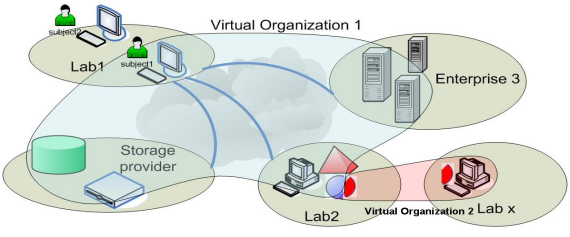


Fig. 1. Multiple partners participate in different VOs

Virtual Organizations may have different attributes as long/short life-time and scopes that vary from intra-enterprise till hundreds of partners. These attributes depend on the goal of a VO and its problem nature. According to its goal, the Virtual Organization creation requires specification of sharing relationships for precise levels of control over how shared resources are put in place and used. This means that the access control policy should precise users' rights at the tapped resources. The granted rights may depend on the context (as time, availability or performance), the user identity and the allocated resource.

3 Environment Characteristics

Having a close look, we point out some issues characterizing the grid environment:

Users: The grid is often build between multiple domains where administrative authorities may add, remove or even change rights of VO users. This is typically the case when an enterprise allocates some employees for a certain project for a certain time, to transfer, after, some of these to another project. Managing the users (without prior knowledge) and their associated rights to access resources at different sites is not a simple task in such a large scale environment.

Resources: VO resources have a volatile nature as the users. A resource may be added or removed or replaced in an environment where faults are frequent. To overcome these problems, resources are often virtualized [16] and then resource description becomes more important than resource naming. Clients need then to

request resources by property ex. capability, quality of service or configuration. Resources properties may be assembled within sets serving as Views for this resource ex. a device may be seen as a computing device and a storage device at the same time.

Reutilization: The use cases of resources cannot be limited whether from the client point of view (data extraction jobs may convert a computing cluster into a high performance data server) or from the provider point of view to respond to various kinds of problems (participation in different VOs) or maximize revenue (pay per usage). Once again resources should not be referenced by name but by properties or a view which is more appropriate to the grid usage.

Multi domain aspect: The grid is founded on the idea of dispersed users and resources within different administrative domains. Centralizing the administration of a VO implies delegating the management of all the users and resources to a single authority. However, a partner in a VO may be a rival in another, so delegating him the control and management of local resources is often not realistic. Each domain needs to manage its resources locally keeping in mind the supplied services' availability. For example, a storage provider may need to move the data on other storage devices locally for maintenance reasons without referring back to the client. An enterprise needs to add locally an employee to a certain workgroup to use the VO resources without returning back to the other partners. Excluding a central authority, the VO requires scalable multi-administration by the involved partners. By multi-administration we don't mean a domain with multiple administrators having the same rights, but rather an environment with different administrators having each partial administration rights within the whole structure. Though the grid considers inter-domain aspects in the Virtual Organization, however it doesn't specify how to put in place such a space.

Delegation: The grid problem definition aims at relieving the user from the burden of resource coordination. To enable coordinated resource sharing, delegation (may span multiple administrative domains) is necessary. Delegation is employed when at runtime a task on a device needs accessing another device on behalf of the user (accessing data, subtasks execution…). Controlling delegation and the delegated rights is necessary not to have access control security breaches.

Contexts: users may be authorized to access resources in certain contexts where outside these contexts access is forbidden. Take for example resources rental for a definite time or even the resources rental within different time periods (ex. from 8h till 15h).

4 Virtual Organization Lifecycle

To better show the context where this work is situated we discuss the major steps of the VO creation process (fig.2) [19]. A goal or a requirement is the motivation for launching the process. A goal may be massive data storage, exploiting certain applications or computing resources utilization. To achieve this goal, required capabilities should be identified. These capabilities are the criteria on which the customer bases

the choice of a service provider. Considering for example the goal to be data storage, the required capabilities may be a certain storage space and performance statistics to be supplied by the provider.

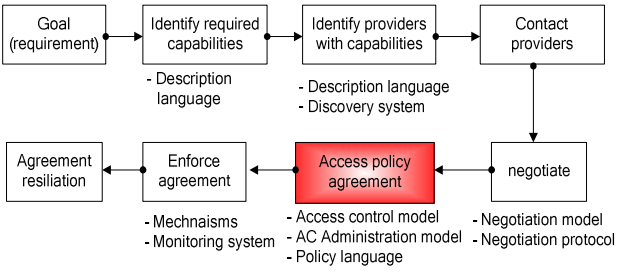


Fig. 2. The VO creation process along with the needs under each step

This necessitates a description language to express a wide variety of parameters. This language may be used by the provider also to describe his services. These descriptions may be published on Internet, where clients may use a discovery system to find and contact the potential providers [16]. We will not detail further this subject in this paper to focus on our main interest the access control policy.

In such a distributed system, customers need to negotiate access to services based on their requirements and the provider capabilities (ex. quality, price, content, type of allocated servers). Thus, the negotiation includes defining users and shared resources as well as specifying the sharing relationships. The process finally closes with the enforcement of the agreement that assures the runtime management of the virtual organization till the end of its lifetime. As existing solutions for access control management within virtual organization we find the Community Authorizations Service (CAS) and Virtual Organization Management System (VOMS). CAS [6] allows enforcing low level user's permission to VO resources as read/write file permissions. This solution was proposed as a mechanism to enforce access control in the Globus toolkit. On the other hand, the Virtual Organization Management System (VOMS) [7], constitutes a server that supplies grid users with X.509 certificates extended with role and group information. The user contacts the server and obtains an X.509 certificate to be used while contacting the resource. GRASP project [8] offers an architecture that uses X.509 certificates to create secure groups (Closed Collaboration Teams) within the lifetime of the Virtual Organization. The VO itself is static, with out-of-band static policy management. These works in the access control domain fall within the "Enforce agreement" block of our life cycle (figure 2). These mechanisms refer to certain access control policy however the works don't go farther to specify this policy and its different actors. The boundaries of a VO are defined by the sharing relationships which are access control policies specifying permissions/prohibition within the VO.

The dynamicity feature of the entities within the virtual organization makes specifying the sharing relationships burdensome. The sharing relationships should be independent of the underlying physical entities by referring to the users function [11][13]

and the resources properties [12]. The role of each partner in producing this policy becomes a part of the distributed access control management [13][14].

5 Access Control Model for VO

The model to be used should take into consideration the Grid environment constraints explained in the above sections. In addition to the contextual policy, the model should represent the different users and resources in such a way that these dynamic entities don't affect the policy validity. This can be done by abstracting the different entities where the management model should attribute to each side the right to manage his users and/or resources (Fig 3). As traditional access models we find the DAC (Discretionary Access Control) [10] with its simple form the access matrix. It gives users rights to do actions on certain objects. However this model neither offers the needed abstraction nor models the different administration tasks. On the other hand, MAC models (Mandatory Access Control) propose centralized management which is not convenient for a distributed multi-administrated environment [10]. Both of these models don't treat contexts. RBAC [11][13] introduces the "role" notion which represents a function in an organization (ex. engineer, administrator). A subject is attributed a role which is associated with a set of permissions (or privileges). Using roles to abstract the subjects responds to the grid needs, however at the "object" side RBAC doesn't offer a similar abstraction. Even though, RBAC doesn't prevent such abstraction, but this abstraction will no more be within the same framework. In addition, RBAC doesn't express contextual permissions or interdictions [12] which are important in the grid environment for example: role "engineer" has the right to use application "appl. 1" for 3 hours.

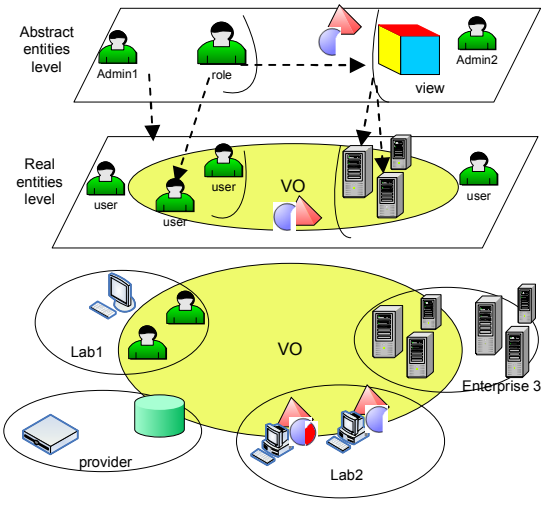


Fig. 3. VO access control relations

DRBAC (distributed role based access control) [9] is a derivative of RBAC model proposed for scalable decentralized access control that spans multiple administrative domains. Even though the distributed management is one of the grid requirements, DRBAC still suffers from RBAC drawback concerning contexts and resources abstraction.

CBAC (Coalition based access control model) [18] is envisaged to treat access control in coalitions where each must be able to share specific resources while ensuring that these resources are safe from inappropriate access. It captures inter-organization relationships and contexts however the administration model is not yet complete to model multiple administrators that each is responsible for a certain part of the system.

6 Or-BAC

In the grid environment each partner has resources that are put in common to be shared by the community. What we need is an access control model that indicates: who can do what in which context. Or-BAC [12] [14] access control model is proposed for modeling a security policy that is not restricted to static permissions and include contextual rules related to permissions, prohibitions and obligations. It aims at introducing an abstraction level that separates access control policy from its implementation.

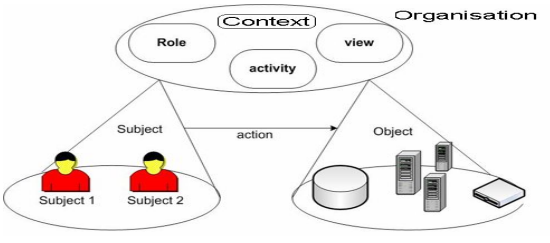


Fig. 4. Or-BAC access control model

OrBAC identifies the “Organization” (figure 4) as an organized group of active entities, i.e. subjects, playing some role or other. Our virtual organization may be modeled as an organization in Or-BAC model since it is the result of a group of entities each playing a role (ex. Provider, consumer, user, resource...) within the whole structure. In this organization, users are abstracted using “Roles” which corresponds to certain privileges in an organization. Abstracting users facilitates user management in a large scale dynamic environment (It is sufficient to assign/revoke a role to a new/departing user). On the other hand, resources also are abstracted using “Views”. A view corresponds to a set of objects that satisfy a common property (storage device, databases, files etc). This notion abstracts the VO resources according to their usage and capabilities. Activities are also used to abstract actions as “read", “write" so it corresponds to a combination of actions.

Finally, the policy is stated in terms of roles, activities, views conditioned by certain temporal or spatial parameters (ex. Secure connection, daytime, time), which are specified as Contexts. Or-BAC with these notions responds to the grid requirements. We may consider a Virtual Organization to be an organization in Or-BAC. Entities in the virtual organization would be roles instead of users, and views instead of objects. The large scale problem of the grid (users, resources) is overcome by the offered abstraction. On the other hand, the multi-administration is also taken into consideration as we will see in the following sections.

6.1 Or-BAC Relationships

There are eight basic sets of entities: Org (organization: an organized group of subjects, playing some role within the group), S (a set of subjects), A (a set of actions), O (a set of objects), R (a set of roles), $\mathcal{A}$ (a set of activities), V (a set of views), C (a set of contexts). Or-BAC considers that org $\subseteq$ S, S $\subseteq$ O. Any entity may have attributes, for instance if S is a subject, then name(S), address(S) represents the name and the address of subject S. Or-BAC also defines relations between these sets:

- Empower is a relation over domains *Org x S x R,* to express that *org* empowers subject *s* in role *r*.
- Use is a relation over domains Org x O x V. *Use (org, o, v)* means that *org* uses object *o* in view *v*.
- Consider is a relation over domains Org x A x $\mathcal{A}$. *which* means that org considers that actions α falls within the activity *a*.
- Define is a relation over domains Org x S x A x O x C, *which* indicates that within organization *org* context *c* holds between subject *s*, action α and object *o*.
- Access control policy is defined by the relation permission. Permission is a relation over domains Org x R x $\mathcal{A}$ x V x C, *Permission (org, r, a, v, c)* means that organization *org* grants role *r* permission to perform activity *a* on view *v* within context *c*.

We consider that the grid Virtual Organization is an organization Org in the Or-BAC context. To construct this space, add users, attribute users to roles and resources to views and specify access permissions, the management model is needed. It should control the activities: management of organizations, management of roles, activities, views and contexts, assignment of users to roles, assignment of permissions to roles, assignment of users to permissions.

6.2 Or-BAC Administration Model (AdOrBAC)

AdOrBAC is the administration model of Or-BAC. It defines special views as URA (user role assignment), PRA (permission role assignment)... Objects belonging to these views have special semantics; they will be respectively interpreted as an assignment of user to role and permission to role. Inserting (deleting) an object in these views enables an authorized entity to respectively assign (revoke) a user to a role and permission to a role. Distributing the administration responsibilities among the grid partners consists of defining which roles are permitted to access the views URA, PRA … or to more specific views when the role has not complete access to one of these views (Fig 5).

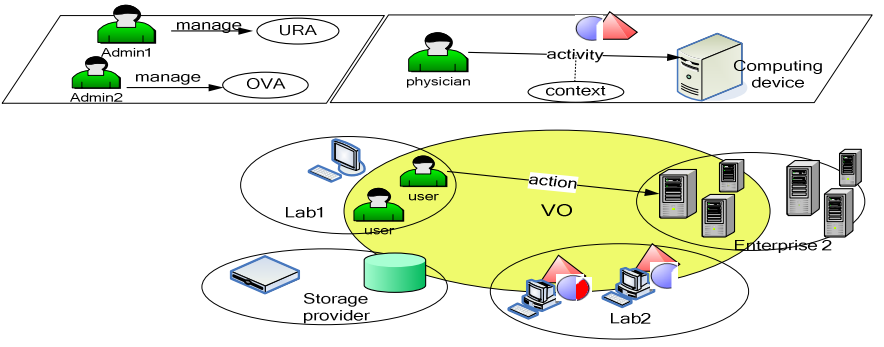


Fig. 5. OrBAC and AdOrBAC model

URA in AdOr-BAC. It is used to determine who is allowed to assign a user to a role and on which conditions. Assigning a user to a role equals adding a new object in a given view called URA (fig 6). Three attributes are associated with this object:

- *Subject* to designate the subject which is related to the assignment.
- *Role* that corresponds to the role to which the subject will be assigned.
- *Org* to represent the organization to which the subject is assigned.

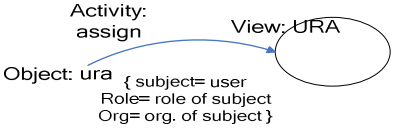


Fig. 6. User role assignment

So to add a user to a special role "physicist" in the physics department "*phy-dept*" at organization "Lab", we have to create a view "URA- physicist - phy –dept" defined as follows:

∀ ura, Use (Lab, ura, URA- physicist - phy -dept) <=> Use (Lab, ura, URA) ^ (role (ura) = physicist) ^ (org (ura) = phy-dept)

This way we defined a view for the physicists in the phy-dept. There is a relationship between adding an object to this view and the relation "empower":

∀ Org, ∀ ura, Use (org, ura, URA) -> Empower (org (ura), subject (ura), role (ura))

"Assign" is the activity to be given to the administrator to do this assignment job.

Permission (Lab, administrator, assign, URA- physicist -phy-dept, default)

This method allows attributing roles administration to a certain administrator whom we grant the permission to assign/revoke users. We may attribute a member of a certain domain the right to add "ura" objects in a view defined of his domain users we may then satisfy the requirement that each domain keeps control on its users.

PRA in AdOr-BAC. Permission role assignment follows the same logic as URA concerning adding an object to a view; however the object has 5 attributes:

Issuer= issuing organization
Grantee, privilege, target = role, activity and view concerned by the permission
Context = designate the context in which the rule can be applied.

There is a link between adding an object to this view and the relation permission:

∀ pra, ∀ org, ∀ context, Use (org, pra, PRA) -> Permission (issuer(pra), grantee(pra), privilege(pra), target(pra), context(pra))

View Object Assignment. In analogy to URA [14], we would like to define a view including certain collection of objects and attribute the manage activity to an administrator (ex. View-admin). Associating a user with the role *View-admin*

Empower (org, user, View-admin)

Attributing permission "manage" to the role "View-admin":

Permission (org, View-admin, manage, VOA-org2, default)

Where *VOA-org2* is the view of a group of resources having an attribute *org* that is equal to org2, it can be defined as follows:

∀voa, Use(org, voa, VOA-org2) <=> Use(org, voa, VOA)^ (org(voa)=org2)

So finally attributing a resource or object to a view in the organization is equivalent to adding an object voa with attributes (object to be added, view to which it should be added, in org as organization) to the VOA-org2 view. The view-admin is in charge of that.

7 Virtual Organization Modeling

Consider the scenario in figure 7 where two organizations need to cooperate forming a Virtual Organization.

Consider the simplified case where organization org2 supplies the resources (provider) and org1 contains the users (consumer). Putting in place the VO needs to define the participating entities from both sides including roles, views, activities and the administration authorities.

Modeling this environment using Or-BAC, we start by the management of the virtual organization itself. Considering that VO is an organization in Or-BAC framework, such an entity may have attributes. As discriminating attributes there are the involved participants, a name discriminating a certain VO within the different VOs that can be set up among the same partners, and may be a certain expiry date after which the cooperation is terminated.

Name (VO) =cooperation 1
Partners (VO) = org1, org2
Time= timelimit

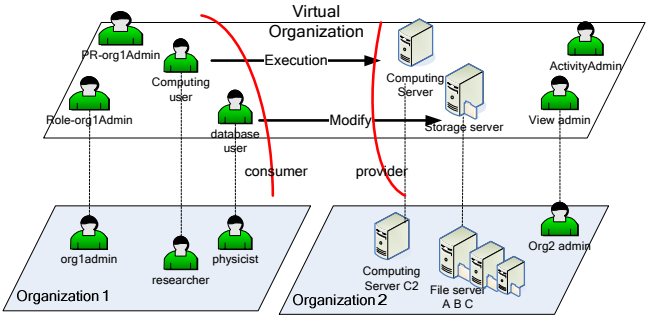


Fig. 7. Or-BAC model in Grid environment

At the administration plan: we should define the relevant roles in the Virtual Organization as:

Relevant-role (VO, Role-org1Admin, PR-org1Admin, ViewAdmin, ActivityAdmin).

In the same way, the relevant views and the Relevant activities names should be defined for the VO. To attribute to a certain "*org1admin*" the role "*Role-org1Admin*" that has the responsibility to *assign/revoke* roles to the users of organization org1: *Empower (VO, org1admin, Role-org1Admin); Permission (VO, roleadmin, manage, URA- org1,default)*

Where URA-org1 is defined as:

$\forall$ *Ura, Use (VO, ura, URA-org1) <=> Use (VO, ura, URA) ^ org (ura) = org1*

To attribute to a certain "*org1admin*" the role "*PR-org1Admin*" that has the responsibility to assign/revoke permissions to roles of organization org1:

Empower (VO, org1admin, PR-org1Admin)
Permission (VO, PR-org1Admin, manage, PRA-org1, default)

Where PRA-org1 is defined as:

$\forall$ *pra, Use (VO, pra, PRA-org1) <=> Use (VO, pra, PRA) ^grantee(pra)∈ {r/ ∃ ura∈ URA-org1, role(ura)=r}^ privilege (pra) ∈ activities(VO)*

To attribute to a certain "org2admin" the role *"ViewAdmin"* that has the responsibility to assign/revoke views to objects of organization org2:

Empower (VO, org2admin, View-org2Admin)
Permission (VO, View-org2Admin, manage, VOA-org2, default)

Where *VOA-org2* is defined as:

$\forall$ *voa, Use (VO, voa, VOA-org2) <=> Use (VO, voa, VOA) ^ org (voa) = org2 ^ Relevant-view(VO ,view(voa))*

To attribute to a certain *"org2admin"* the role *"ActivityAdmin"* that has the responsibility to assign/revoke activities to actions comprehensible by organization org2 and its different resources:

Empower (VO, org2admin, ActivityAdmin)
Permission (VO, ActivityAdmin, manage, AaA-org2, default)

Where *AaA-org2* is defined as:

$\forall aaa, Use (VO, aaa, AaA\text{-}org2) <=> Use (VO, aaa, AaA) \wedge org (aaa) = org2$

Assigning concrete level to the abstract level: according to figure 5 authorized roles/users Role-org1Admin has the permission to empower a user with a role as in the following rules:

Empower(VO, researcher, computinguser)
Empower(VO, physicist, databaseuser)

ViewAdmin has the permission to use an object in a view as in the following rules:

Use(VO,storageserver,FileserverA&FileserverB&FileserverC)
Use(VO,computing server, computingserverC2)

ActivityAdmin has the right to consider an action within an activity as in the following rules:

Consider (VO, execute, Execution)
Consider (VO, write, Update)
Consider (VO, read&write, Modify)

PR-org1Admin has the right to permit a role to do an activity on a view as following:

Permission(VO, database user, Modify, storage server, workTime)
Permission(VO, computing user, Execution, computing server, day&night)

This way using Or-BAC we modeled our grid environment abstracting the users and resources within a single framework. This framework allows different partners to participate in specifying the access control policy keeping the autonomy and flexibility in managing their local assets. This formalism should be employed within a cooperation and negotiation framework to put in place a multi administrated VO.

Delegation: Modeling delegation is not shown yet. We will pass in general to see how is the demarche; however we will not enter in details being limited in this paper. Delegation in the grid context is to give some device working on behalf of a user some rights to enable it access other resources. This set of rights is normally a subset of the set of rights of the user himself. Delegation may be supported by the model when the internal policy of an organization authorizes giving the administrator role the right to assign roles for subjects in other domains for a special task. So to model delegation there should be a view containing the delegated entities and the right to give them roles or sub roles in order to complete the assigned task. This way we can control delegation which depends on the policy of the client as well as the provider. The client may sometimes not trust the object and thus doesn't delegate it. The same can be said about the provider who may not accept that his resources go farther to propagate anonymous malicious activities for example (Distributed Denial of Service attacks).

8 Modeling Issues

It is necessary to refer back to the set of constraints defined above in section 3 and relate them to the model terms. Abstraction of entities at the VO level allows having two distinct levels: valid policy and dynamic entities. On the other hand, every partner has his settled internal access control system. It is not desired to modify this system (users, roles, privileges…) each time the partner adheres to a different virtual organization. Though the activities of user in the virtual organization may reveal the hierarchy and the different responsibilities between the involved users, however this should not compromise the internal organization structure. The different privileges given to users at the virtual organization level should be subject to the intra-organization management. The user role assignment policy is an internal issue according to which the organization attributes a role to a user.

The same discussion can be employed to argue that the OVA policy also should be internal to the provider domain. However this internal policy should take into consideration the commitments of the provider to supply a specific service with particular characteristics. Following this logic we attribute the URA to the organization of the user and the VOA to the organization under which the objects are administered. We base on a pre-virtual organization negotiation to give certain roles (ex. Role admin, View admin) certain privileges on certain resources within certain contexts. These contexts may depend on specific partner parameters (resource quality of service, rental time, and specific periods of time…).

9 Conclusion and Future Works

The grid has a goal enabling coordinated resource sharing and problem solving in dynamic multi-domain virtual organization. For this reason it sets up a virtual organization which constitutes the shared common space between the different partners. However the grid doesn't indicate how to construct such a space in a dynamic, multi-administrated environment. On the way to dynamically create virtual organizations we had to find a model that organizes the different responsibilities in a VO. It should take into consideration the environment constraints as multi-organizational aspect, large scale, dynamic resource allocation… We chose Or-BAC for this mission which shows flexibility to model these situations. Or-BAC models a multi-administered environment using the "role view activity" abstraction which abides to the other constraints (large scale and dynamic resources). With this abstraction Or-BAC policy rules are independent of the physical underlying infrastructure. What is still need to be detailed in our model is the delegation aspect which is necessary for the domain of ubiquitous computing. We will treat this issue independently [15].

The next step after modeling is to implement the automated creation of such environment by discovery, negotiation, and generation of an SLA-like policy [8] which may be powered with QoS parameters along with the access control ones. Finally this policy needs enforcement using appropriate mechanisms within the grid layers [5].

References

1. B. Nasser, A. Benzekri, R. Laborde, F. Grasset, F. Barrère. "Access Control Model for Grid Virtual Organizations", to appear in ICEIS conference, 2005.
2. Grid Support for Ubiquitous Computing Research Group Global Grid Forum. http://ubigrid.lancs.ac.uk/ubicomp_rg_ charter.html
3. Gilles Fedak, Cecile Germain, Vincent Neri, and Franck Cappello, 2001. XtremWeb: A Generic Global Computing System. CCGRID2001, workshop on Global Computing on Personal Devices, May 2001, IEEE Press.
4. Arcot Rajasekar, Michael Wan, Reagan Moore, Wayne Schroeder, George Kremenek, Arun Jagatheesan, Charles Cowart, Bing Zhu, Sheau-Yen Chen, Roman Olschanowsky. Storage Resource Broker – Managing Distributed Data in a Grid. Available online: http://www.npaci.edu/DICE/Pubs/CSI-paper-sent.doc
5. Ian Foster, Carl Kesselman, Steven Tuecke, 2001. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. In, International J. Supercomputer Applications,15(3),2001.
6. Cannon S., Chan S., Olson D., Tull C., Welch V., Pearlman L., 2003. Using CAS to manage Role based VO sub-groups. In CHEP 2003, La Jolla, California, available online: http://www.globus.org/security/CAS/Papers/CAS-group-CHEP03.pdf
7. Alfieri R., Cecchini R., Ciaschini V., dell'Agnello L., Frohner A., Gianoli A., L˝orentey K., and Spataro F., 2003. "VOMS, an authorization system for virtual organizations", DataGrid Project, available online: http://grid-auth.infn.it/docs/VOMS-Santiago.pdf
8. Djordjevic I., Dimitrakos T., Phillips C. An architecture for dynamic security perimeters of virtual collaborative networks. Proc. 9th IEEE/IFIP Network Operations and Management Symposium, (NOMS 2004), April 2004. IEEE-CS.
9. Eric Freudenthal, Tracy Pesin, Lawrence Port, Edward Keenan, and Vijay Karamcheti. dRBAC: Distributed Role-based Access Control for Dynamic Coalition Environments
10. Pierangela Samarati, Sabrina De Capitani di Vimercati. Access Control: Policies, Models, and Mechanisms.
11. Ravi Sandhu, Edward Coyne, Hal Feinstein, Charles Youman, 1996. Role-Based Access Control Models. IEEE Computer, vol. 29, n° 2, pp.38-47, février, 1996.
12. Anas Abou El Kalam, Rania El Baida, Philippe Balbiani, Salem Benferhat, Frederic Cuppens, Yves Deswartes, Alexandre Miege, Claire Saurel, Gilles Trouessin , "Organization Based Access Control" , available online: http://www.rennes.enst-bretagne.fr/~fcuppens/articles/Or-BAC.pdf
13. Sandhu R., Munawer Q., 1999. The ARBAC99 Model for Administration of Roles. In Proceeding of the 15th Annual Computer Security Applications Conference (ACSAC'99), Phoenix, Arizona, 6-10 December 1999, IEEE Computer Society, pp. 229-241.
14. Frederic Cuppens, Alexandre Miege. Ad-ORBAC: An Administration Model forOr-BAC. Available online: http://www.rennes.enst-bretagne.fr/~fcuppens/articles/csse04.pdf
15. Welch, V., Foster, I., Kesselman, C., Mulmo, O., Pearlman, L., Tuecke, S., Gawor, J., Meder, S. and Siebenlist, F. (2004). X.509 proxy certificate for dynamic delegation, Proceedings of the 3rd Annual PKI R&D Workshop.
16. Karl Czajkowski, Ian Foster, Carl Kesselman, Volker Sander, Steven Tuecke, 2002. "SNAP: A Protocol for Negotiation of Service Level Agreements and Coordinated Resource Management in Distributed Systems". Draft submission to JSSPP'02 April 30, 2002. Available online: http://www-unix.mcs.anl.gov/~schopf/ggf-sched/GGF5/sched-GRAAP.3.pdf

17. Ian Foster, Carl Kesselman, 1997. Globus: A Metacomputing Infrastructure Toolkit. Intl J. Supercomputer Applications, 11(2):115-128.
18. E. Cohen,R. Thomas, W. Winsborough, D. Shands. Models for coalition based access control (CBAC).
19. Nitin Nayak, Tian Chao, Jenny Li, Joris Mihaeli, Raja Das, Annap Derebail, Jeff Soo Hoo, "Role of Technology in Enabling Dynamic Virtual Enterprises". Available online: http://cersi.luiss.it/oesseo2001/papers/13.pdf

Interoperability Supported by Enterprise Modelling

Frank Walter Jaekel[1], Nicolas Perry[2], Cristina Campos[3], Kai Mertins[1], and Ricardo Chalmeta[3]

[1] Fraunhofer IPK, Pascal Str. 8-9, 10587 Berlin, Germany
{Frank-Walter.Jaekel, kai.mertins}@ipk.fhg.de
[2] Ecole Centrale de Nantes, IRCCyN, 1 rue de la no, 44321 Nantes, France
Nicolas.Perry@irccyn.ec-nantes.fr
[3] Grupo de investigación en Integración y Re-Ingeniería de Sistemas (IRIS), Dep. de Llenguatges i Sistemes Informàtics, Universitat Jaume I, 12071 Castelló, Spain
{camposc, rchalmet}@uji.es

Abstract. The application of enterprise modelling supports the common understanding of the enterprise business processes in the company and across companies. To assure a correct cooperation between two or more entities it is mandatory to build an appropriate model of them. This can lead to a stronger amplification of all the cross-interface activities between the entities. Enterprise models illustrate the organisational business aspects as a prerequisite for the successful technical integration of IT systems or their configurations. If an IT system is not accepted because its usefulness is not transparent to the staff members, then it quickly loses its value due to erroneous or incomplete input and insufficient maintenance. This at the end results in investment losses.

The paper exemplifies the strengths, values, limitations and gaps of the application of enterprise modelling to support interoperability between companies. It illustrates a proposal for a common enterprise-modelling framework. This framework is presented in terms of problems to face and knowledge based methodological approach to help solving them. A specific application demonstrates enterprise modelling and the synchronisation between the models as prerequisite for the successful design of Virtual Enterprises.

1 Introduction

The implementation of information systems and new organisational structures into and between companies requires discussions between different stakeholders of the enterprise (e.g. process design experts, managers, process owners, IT experts). Therefore the modelling of enterprise processes including related information systems and organisational units is an essential step in the process of changing and improving enterprise structures. The target is to achieve a common understanding of requirements of a new system. This is true for big companies as well as for small and medium size enterprises (SME). Furthermore the enterprise modelling bridges the gap in transforming process organisation of an enterprise

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 552–561, 2005.
© Springer-Verlag Berlin Heidelberg 2005

and the processes implemented within the IT systems. The complexity of the modelling approach increases if used across enterprise networks.

The modelling of enterprise business processes is growingly becoming a well known technique especially within big companies. Now also SMEs are forced by their customers to increase the transparency of their processes. Moreover, the need of IT support such as ERP systems increases for SMEs. The establishment of information systems within a company is a difficult task. Various applications of IT systems are not efficient because of a lacking acceptance on the user side and of deficits between the real process organisation of the enterprise and the support of the IT systems.

Experiences from industrial projects illustrate that companies which buy an IT system without a clear strategy for enterprise business process improvement and little knowledge regarding the organisational effects often fail in applying the software. Therefore, big companies as well as SMEs require a modelling approach to create an enterprise (business) process blue print for a successful implementation of IT systems. The model is oriented to and across process owners or stakeholders of operational and management departments. Consequently the description of the process structure and its relations to different resources such as organisational units, IT infrastructure, information exchange, etc. has to be easily understandable.

Enterprise modelling concerns awareness of enterprise cultural particularities. The goal is to answer the question: 'How to make different enterprise models interoperable made from different modelling methodologies languages and metamodels, modelling background and environment?'.

The topic is covered by the European INTEROP [1] Network of Excellence concerned with inter-operability research for networked enterprises applications and software, its goals, rationale and early results and by the European integrated project ATHENA [2](Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Applications).

To address a Common Enterprise Modelling Framework definition, the first step is to establish a common base of understanding of different modelling constructs across different modelling languages that is a common modelling language such as the INTEROP UEML approach. The second step is to take into account the different ways of representing the real world within the model content, including aspects such as cultural and regional differences both in enterprise way of working but also in the way to build models, different objectives driven models and so on.

2 Models Across Organizations

2.1 Problems of Enterprise Modelling Between Companies

In the actual situation regarding enterprise modelling several modelling methods and tools are used in enterprises (Figure 1). For example, MO^{2}GO [3] supporting the integrated enterprise modelling is preferred because of a fast and easy understandable modelling method across different stakeholders. GRAI [4]Tools

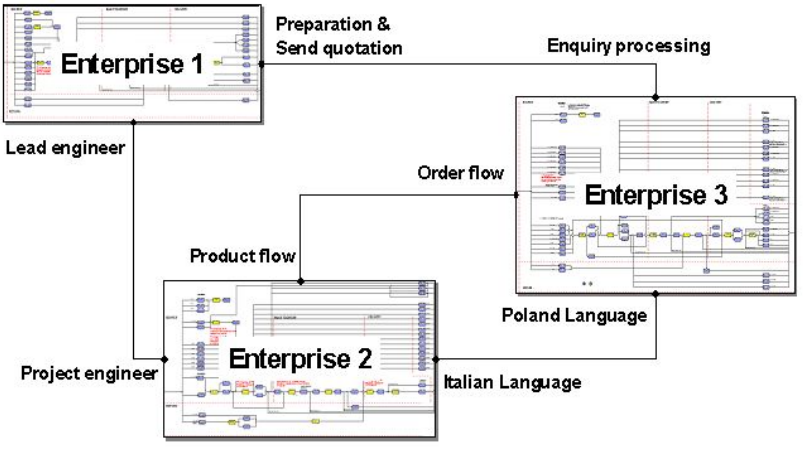


Fig. 1. Methods and Tools

are prioritised especially for modelling the decisional processes of an enterprise. ARIS [5] is popular for enterprise modelling especially in the IT domain and IT departments. METIS [6] supports a very flexible meta modelling and therefore a good adaptation of the user wishes according the modelling constructs. The information covered by these tools is similar. Therefore, to save the investment for method training and model elaboration an exchange of the information modelled within the different tools should be provided. In the first step within INTEROP this is a topic of the Unified Enterprise Modelling Language (UEML) [7].

The problem of dealing with different models does not only depend on the modelling language. The same issue might be defined in two different models with different terms (e.g. Lead engineer / Project engineer) but at the same time in a third model these terms may have another meaning. Instead of the modelling language the natural language might be hindered by an information exchange and cooperated work on the models because a translation into an interlingua e.g. English might result in misinterpretations without having a common ontology support. The perspective between two models dealing with the same information might be different e.g. order processing or product processing concerning the external interfaces of an enterprise. The structuring of the processes as well as the design of the process chains might be dissimilar e.g. the two processes 'Preparation' and 'Send quotation' could appear in another model just the process 'Enquiry processing' (Figure 2). These are some examples of the problems under consideration in the INTEROP work around synchronisation of distributed enterprise models. Further problems arise concerning the management of such distributed enterprise models. An enterprise model associated with different other models requires clear procedures of how to perform changes [8].

The intension of the paper is to motivate enterprise modelling supporting interoperability. More information regarding requirements and state of the art of enterprise modelling in the context of interoperability can be found in [9] [10] [11].

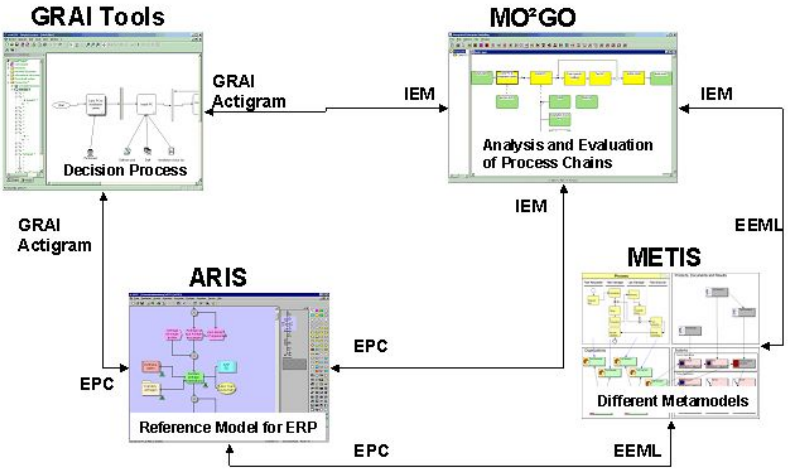


Fig. 2. Same modelling language but different understandings

2.2 Values of Enterprise Modelling

The application of enterprise modelling supports the common understanding of the enterprise business processes in the company and across companies. The company is supported to succeed in reducing the throughput times, in improving the process quality, in reducing costs and therefore in improving the customer satisfaction and competitiveness. Enterprise modelling should be the basis of the information system planning process. The use of enterprise modelling for supporting and achieving company interoperability has different motivations, for example:

- Identification of companies' potentials for acting within different cooperation.
- Enabling companies to participate within collaborations through gathering the required data from the companies.
- Clarification of the connection between the operating processes of the companies and the required IT support through the illustration of additional operations.
- Model supported coordination, composition and synchronisation of organisation structures and business processes between the companies, especially identification of the aspects which support or, what is even more important, inhibit interoperability.

3 Distributed Project Management and Models

Lean extended enterprise and build-to-order induce better integration of Product Lifecycle Management activities that go through computer-aided systems and knowledge-based information environments. This change of landmarks from

physical documents to electronics claims to redefine information support functionalities and knowledge-based tools to support worldwide project collaboration. Moreover, extended enterprises and collaborative projects have to analyse and control their core competencies to react efficiently to the market challenges (time to market, variability of products..).

The number of different enterprise concepts and complexities caused by different interpretations of these concepts encourages enterprises to standardise concepts and formalise behaviour. These efforts build re-usable and adaptable platforms and imply deep business architecture redeployments. The rapidly changing environment requires convenient collaboration and knowledge integration tools, and interoperability between different information sources. As a result, Knowledge Management has become a key facilitator in improving the global competitiveness of companies. As competitive drivers are forcing companies to innovate and change, effective knowledge management is essential to realise and also efficiently implement these changes.

To reach these goals, project management have both to integrate this expert's knowledge and the collaborative project constrains. Consequently interoperability, in terms of synchronisation problems, will occur at these two levels. We will mainly describe the first aspect assuming that a same approach can be applied for project models and alignment.

3.1 Experience Feed Back and Method Proposal

Based on experimental cases we propose knowledge based projects methodology with three phases to optimise and ensure coherent enterprise project management:

- Project infrastructure definition, defines the fundamental elements of the project (domain infrastructure). Based on a syntax/semantic approach of the project's concepts model in order to differentiate concepts. The analysis of their relationship makes the domain architecture. Concepts 'behave' differently according to their context.
- Project architecture, explains the relationships between these elements and the way they are deployed (TO BE situation) in order to measure their efficiency.
- Document generation, describing the knowledge-based application specification for the software developer.

3.2 Projects

Next the two projects that integrate this methodology are described.

Information Consolidation Tool in Order to Build Structured Knowledge-Based Information Environments. This is a French National Project [12], including 5 academic partners, an aircraft manufacturer (the client) and a CAD-CAM System developer, in which we experimented different aspects

of the distributed model management. Clear process definition, steps deliverables and objectives, concepts and working area had to be expressed. Based on the knowledge, experience and requeriments of the end user, the academic works had to capture and formalize the knowledge, improve and strengthen it before specifying software functionalities. We aimed to address 'efficiency of experts to share knowledge', and, due to our project team structure, we had to experiment it between partners. Indeed, specifications or constraints are usually transmitted from one expert to the other in a global convergence. The differences between their competencies limit the global understanding of problems. Computer integration in the expertise chain aims to optimise this kind of relations and thus the use of enterprise knowledge. The main difficulty encountered is to control the complexity of information quantity and informality. A harmonisation of the work-structure will reinforce the efficient use and clearness of information.

Economical Model Integration. Software developers want to unify the four software solutions they developed independently [13]. The purpose is the cost estimation (costing) or sale price determination (pricing) in the micro-electronics field. The goal is to describe a generic economic model used for the determination of the product industrial value. Each tool answers various aspects from the silicon wafer to the finished products (electronic boards, mobile phone). The expert, distributed in different structures (production plant, design offices, buyer or seller services) had to unify parts of their costs models, their calculations rules, data inputs. The complexity relies on the fact that at different phases of the product life cycle, these software are used either by an engineer, a seller or a buyer. Consequently, their own integration of the tools in their project management is very different (objectives, information truth).

In both cases, we used a project modelling methodology supported by the MOKA supporting tool PC-Pack to build common ontology from expert documentation, the modelling tool MEGA, to perform Project Knowledge mapping with a systematic exploration, UML-like activity diagrams and UML class diagrams and time model synchronisation have been used as a possible support of these models alignment. The interest of these formalisms is that they propose a strong common syntax but let people free to rebuild their own semantic interpretation of meta-models thereby ensuring the whole project evolution in coherency with the initials objectives.

In term of model synchronisation, these two examples illustrated different distributed knowledge based projects, integrating cultural, geographical differences between partners or co-contractors. The spaces / domains comparison helps to analyse the model compliance (including model coherency, bijection, mapping of concepts, Meta model definition) and determine the synchronisation needs (this is a part of the domain infrastructure definition).

The use of a common methodology based on six core concepts (Syntax/ Semantic: to give shareable modelled concepts, Infrastructure/Architecture: to define concepts and their interrelationships, Domain/Project: to represent 'AS IS' and 'TO BE' situation) helps the partners to understand each other and share models, information and requirements to perform the works. It still remains the

problem of the life of such a distributed project model, for instance how to integrate new final user requirements on the knowledge based tool that rely on different fields of expertise after the end of the specification phase, or how to manage the evolution of the initial project objectives (financial support reduction and technical adaptations for example).

4 Enterprise Modelling and Virtual Enterprises

A Virtual Enterprise is a temporary alliance of independent enterprises that come together to share resources, skills and costs, with the support of the Information and Communication Technologies, in order to better attend market opportunities. To design an efficient and flexible Virtual Enterprise that gives the appearance of being a single enterprise to customers is a very complex task [14]. A Virtual Enterprise involves a great number of organizations, usually SME, that need to closely collaborate and to be in contact in order to achieve their objectives (competitiveness, better service for their clients, etc.) in this context the use of enterprise models is a key factor became successful.

4.1 Methodology for Developing a Virtual Enterprise

In order to help the creation and management of a Virtual Enterprise, partners develop models using different Enterprise Modelling Languages and different background knowledge. These enterprise models need to be interchangeable and understandable for people involved in each enterprise and for the whole virtual enterprise. In addition, Virtual Enterprises need to update their models due to the natural evolution of business, new legal requirements, changes in the strategy of the partners, and so forth. This kind of changes can affect concepts, business, results and other aspects in enterprise models that are needed to work correctly in real time [14]. Therefore synchronisation is really important in the process of setting up a Virtual Enterprise and it becomes more critical when a Virtual Enterprise is actually running.

The methodology for the virtual enterprise integration developed by IRIS group [15] shows how to set up practices and procedures in order to integrate a virtual enterprise. This methodology proposes (1) the definition of the conceptual aspects of the virtual enterprise and of each single enterprise (mission, vision, strategy, politics, and enterprise values); (2) the redesign of the new process map (internal business processes and cross-organisational business processes that are affected by changes), according to the previously defined concepts; (3) implementation of the VE new process map.; and (4) the extension of the information system (and the technological infrastructure) to support the process map of the virtual enterprise, considering the different levels of decision and the support technology. This methodology has been applied in several projects where thee adequate use of model languages and synchronisation between these models have been critical aspects to be successful.

These four points defined by the methodology are highly supported by the use of enterprise models, and consequently members in a virtual enterprise need to

collaborate and to make al their enterprise models interchangeable and moreover, interoperable and synchronized. For the successful working on and integration of a Virtual Enterprises model contents and models management is a key issue to deal with. Therefore, it is necessary to establish the condition and the bases for good enterprise enterprise model management and synchronization.

4.2 Needs for Enterprise Modelling in Virtual Enterprises

The synchronization and management of enterprise models in a Virtual Enterprise is required under several aspects (e.g. model data, responsibilities, motivation, knowledge, configuration, social aspects etc.). The connection of different and distinct enterprise models is not limited to 'technical' or modelling language problems because, for instance, the same 'enterprise process' modelled with the same modelling language under the same objectives and requirements may differ whenever modelled by different persons.

As it is mentioned in this paper several modelling methods and tools are used in enterprises, and the application of these tools and models helps to common understanding. The same necessities analysed for enterprise interoperability can be found in the integration of a Virtual Enterprise and the same solutions must be taken into account. The application of the methodology developed by IRIS group demonstrates that the correct use of models is essential integration of a Virtual Enterprise.

Clear procedures stating how to manage and synchronize such models, templates and reference models, use of reference ontologies, training about models use and guidelines for modelling processes will help to increase the acceptance, use and results of enterprise models in the developing and management of a Virtual Enterprise and consequently improves the interoperability of all of its partners.

5 Conclusions

The reflections above illustrate the need of enterprise modelling to achieve and support interoperability between organisations. The INTEROP approach of the synchronisation and management of distributed enterprise models focuses on the organisational aspects of such models e.g. the common understanding regarding modelling structures or terms given to modelling elements to express the model content [7]. The INTEROP reference cases [11]illustrate the advantage, needs and requirements for enterprise modelling regarding interoperability as well. Most of the reference casess started with a modelling phase introducing different tools and methodologies. It also illustrates that there is not any general procedure applied and the models depend on how the organisation provides the modelling activities. Moreover, the involvement of the enterprise stakeholders is different. Under these circumstances one can imagine the problem of a company (e.g. a SME supplier) which has to participate into different co-operations (e.g.Virtual Enterprises) and has to be compliant with the other enterprise models. First of all, the modelling language might be considered. The

solution could be using the INTEROP unified enterprise modelling language (UEML) approach [16] [17]. But afterwards the content of the model needs to be related to other models (structures, terms, etc.). Moreover, for achieving both compatibilities in the language and in the content (modelled information), the management of the decentralised models is required. What about changes within the model of the SME? Should they be reflected, directly, in all network models in which the SME participates? What are the results and implications of such changes?

Organizations develop models using different languages and different background knowledge. In order to achieve enterprise interoperability, it is necessary that these models will be interchangeable and comprehensible for people involved in the organization processes. The possibility of different companies to cooperate generates the necessity for models to be connected in a dynamic way. Changes in one of the models of an enterprise can affect processes, decisions and important aspects on the side of other partners. Therefore, synchronization is necessary among models from different enterprises in order to deal with changes, evolution and different views. This is a critical aspect (when models represent enterprise processes, information, organizational structures, products or decisions) for those who are closely connected to the same supply chain or to extended or virtual enterprises. methods have to be elaborated in order to support the model synchronisation and the decentralised usage of these models.

Acknowledgments

The presented research activities are partially supported by the European Network of Excellence IST-508011 'Interoperability Research for Networked Enterprises Applications and Software' INTEROP [1] and by CICYT DPI2003-02515.

References

1. INTEROP: Interoperability Research for Networked Enterprises Applications and Software NoE (IST-2003-508011). http://www.interop-noe.org (2005)
2. ATHENA: Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Applications) Project (IST-2003-2004). http://www.athena-ip.org (2005)
3. MO^{2}GO: Website. http://www.moogo.org (2005)
4. GRAITools: Website. http://www.graisoft.com (2005)
5. ARIS: Architecture of Integrated Information Systems. http://www.aris.org (2005)
6. Metis: Website. http://www.trouxmetis.com (2005)
7. Berio, G., Mertins, K., Jaekel, F.W.: Common Enterprise Modelling Framework for Distributed Organisations. In: 16th IFAC World Congress, Prague (Czech Republic) (2005)
8. SPIDER-WIN: Supply Information Dynamic Exchange and ContRol by Web-based Interaction Network. http://www.spider-win.de/spider-win.htm (2005)
9. Knothe, T., et al.: UEML Public Deliverable D2.3. http://www.ueml.org (2003)

10. Garcia, A.B.: ATHENA Public Deliverable D.A1.1.1 Enterprise Modelling Techniques and Technologies to Support Enterprise Interoperability. http://www.athena-ip.org (2004)
11. Jaekel, F.W., Piddington, C., Doumeingts., G.: INTEROP Public Deliverable D12.2 Deliverable 3.1. Reports on the Pilot Implementations and on the Possibility to Generalise the Methodology to Develop Take-up Actions towards SME's. http://www.interop-noe.org (2005)
12. Candlot, A., Perry, N., Bernard, A., Khodja, S.A.: Deployment of an Innovative Resource Choice Method for Process Planning. In: CIRP International Seminar on Manufacturing Systems, Florianapolis, (Brasil) (2005)
13. Perry, N., Mauchand, M., Bernard, A.: Integration of Cost Models in Design and Manufacturing. In: Advances in Design, Springer Series in Advanced Manufacturing, H. ElMaraghy and W. ElMaraghy (2005)
14. Chalmeta, R., Campos, C., Grangel, R.: References architectures for enterprises integration. The Journal of Systems and Software **57** (2001) 175–191 Elsevier.
15. Chalmeta, R., Grangel, R.: ARDIN Extension for Virtual Enterprise Integration. The Journal of Systems and Software (2003) Elsevier.
16. Berio, G., et al.: Deliverable 3.1. the UEML Project. http://www.ueml.org (2005)
17. Berio, G., Anaya, V., Ortiz, V.: Supporting Enterprise Integration Through a Unified Enterprise Modelling Language. In: EMOI Workshop, Riga, J. Grundspenkis and M. Kirikova (2004)

Using Ontologies for XML Data Cleaning

Diego Milano, Monica Scannapieco, and Tiziana Catarci

Dipartimento di Informatica e Sistemistica,
Universitá degli Studi di Roma "La Sapienza", Via Salaria 113, Roma, Italy
{milano, monscan, catarci}@dis.uniroma1.it

Abstract. Real data is often affected by errors and inconsistencies. Many of them depend on the fact that schemas cannot represent a sufficiently wide range of constraints. Data cleaning is the process of identifying and possibly correcting data quality problems that affect the data. Cleaning data requires to gather knowledge on the domain to which the data refer. Anyway, existing data cleaning techniques still access this knowledge as a fragmented collection of heterogenous rules and ad hoc data transformations. Furthermore, data cleaning methodologies for an important class of data based on the semistructured XML data model have not yet been proposed. In this paper we introduce the **OXC** framework, that offers a methodology for XML data cleaning based on a uniform representation of domain knowledge through an ontology We describe how to define XML related data quality metrics based on our domain knowledge representation, and give a definition of various metrics related to the *completeness* data quality dimension.

1 Introduction

Real data is often affected by errors and inconsistencies. Errors can be introduced by users during data entry, or depend on applications' behavior. Typical data management systems have the possibility to avoid certain problems. For example, relational DBMSs can perform various checks at data entry or when certain operations are performed. XML documents used for data exchange can be validated against a DTD or other kind of schema. Many problems could be avoided if the used DBMS was able to actually enforce all the constraints that must hold on the data in order for them to be error-free and consistent. Anyway, DBMS and applications managing XML documents often fail at enforcing such a wide range of constraints. This may be due to limitations of the data models used to represent data or to data management systems. Furthermore, it can be due to bad design in the initial modeling phase or from design choices motivated by efficiency reasons.

This is particularly true for an important class of data, that of data modeled as XML documents, as a widely diffused schema formalism available for XML,i.e. DTD, is particularly weak at expressing some constraints that are essential to ensure the quality of XML data. Furthermore, while long-time established good design methodologies exist for the relational case, the problem of defining guidelines for XML schema design has been only recently addressed, e.g.[1,8], and no consolidated methodology is available yet.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 562–571, 2005.
© Springer-Verlag Berlin Heidelberg 2005

The presence of errors and inconsistencies may have serious consequences, including data corruption or loss and misusage of data, and may eventually lead to sensible losses of money for enterprises, e.g. in terms of missed orders, loss of image and so on. In order to define and quantify the errors that affect the data, some definitions of *data quality* have been recently introduced, e.g [4,12,5]. Data quality is a young research field, and still lacks a commonly agreed set of definitions. Anyway, it is commonly accepted that the quality of data is a complex concept, that can be analyzed under various *data quality dimensions*. Various sets of such dimensions have been proposed in the literature. Some of them are highly domain specific, but it is possible to identify a common subset of such dimensions describing essential data quality features. These dimensions are *accuracy*, *completeness*, *consistency* and *currency*.

Data quality has been studied so far at quite an abstract level, and sometimes specialized to the case of relational data. Quality related issues for XML documents are considered in some works, e.g. [3,13]. However, the problem of defining appropriate metrics to measure the quality of XML data has not yet been considered.

Data cleaning is the process of identifying and possibly correcting data quality problems that affect the data. Cleaning usually requires gathering knowledge about the properties of data to be cleaned and the various constraints characterizing the domain they represent. Unfortunately, existing techniques still treat this knowledge as an heterogeneous set of variously represented rules and ad hoc data transformations. This forces an unnecessary fragmentation of the cleaning process, that requires frequent backtracking after a transformation has been applied to data, as new transformations may lead to data which is inconsistent with the constraints enforced by the previous ones. Also, it makes harder to generate cleaning transformations in automatic or semiautomatic way.

In our opinion, representing all knowledge on the application domain in a uniform way is a fundamental problem. In order to solve it, it is necessary to adopt a modeling formalism which is expressive enough to allow for the representation of all the details of such knowledge, and makes it available in a way that makes easier to automatize cleaning tasks. As a solution to this problem, we propose to represent such knowledge through an ontology representation language. The language should be expressive enough to allow modelling the application domain in a form much richer than that given by the data schema alone, and have formal semantics in order to allow automated reasoning over the ontology.

A few frameworks have been proposed to address the cleaning task in the relational case (e.g. [2], see also [9,11]), based on declarative languages to specify transformations. Such works, however, don't deal with the problem of representing domain knowledge in a uniform way. Furthermore, to the best our knowledge, no approach has been proposed yet to address cleaning of XML documents.

In this paper, we propose a framework for data cleaning of XML documents called the **OXC** (**Ontology-based XML Cleaning**) framework, in which all the knowledge gathered through a domain analysis activity (e.g. performed by a domain expert) and from the DTDs of the documents to be cleaned is represented

through a unified, expressive modelling language as a *reference ontology*. On the base of this ontology and of a mapping between the DTD and the ontology itself, we define appropriate quality metrics for XML documents. The **OXC** framework comprises *(i)* a methodology for data quality assessment and cleaning based on the reference ontology and *(ii)* an architecture for XML data cleaning based on such methodology. In this paper we will describe the **OXC** methodology. Details of the design of the **OXC** architecture can be found in [6].

The rest of this paper is organized as follows. In Section 2 we introduce our methodology for XML data cleaning. Section 3 formalizes the problem of defining quality metrics based on the reference ontology, and defines a set of metrics related to the *completeness* data quality dimension. Section 4 gives some conclusions and illustrates future directions of our work.

2 Methodology

Data cleaning needs to rely on a richer data representation than that provided by schema representation languages like DTD and XML Schema. Such representations should be expressive enough to capture additional knowledge, for example additional constraints, beside those already expressed by a schema available for the data to be cleaned. Furthermore, it should have formal semantics in order to allow automated reasoning on the knowledge it expresses. In this section, we define a methodology that allows to clean XML data exploiting a formal ontology-based representation of the knowledge contained in a DTD, with respect to which the examined XML document is valid, combined with additional semantic constraints specified by a domain expert. We call the methodology **Ontology-based XML Cleaning** (**OXC**) methodology.

The **OXC** methodology is applied to an XML document, and a DTD to which it conforms. An ontology capturing domain knowledge is first designed by a domain expert. Such domain knowledge includes any knowledge already expressed in the DTD, and adds to it any additional knowledge that should appear in a good conceptual design of the considered domain, but has been left aside (because of limits in the schema language or because of bad design) when actually designing the DTD. As a parallel step to that of designing the ontology, a *mapping* relation between the ontology and the DTD is also defined. The ontology (together with the mapping) gives a *reference world* against which to measure the quality of the XML document. Notice that the given XML document is valid with respect to the initial DTD, but might be "dirty" with respect to this new, enhanced representation. The mapping is used in order to : (i) define data quality dimensions (and hence their violations) with respect to the ontology; (ii) perform data quality improvement by relying on the semantics encoded by the ontology.

The final result of the **OXC** methodology is a *cleaned* XML document which is not simply valid with respect to the initial DTD, but it is *ontology-valid*, that is all quality checks, defined in terms of the ontology, have been run on it, and possible correction actions have been engaged on the document as well.

In the following sections we don't deal with the actual creation of this reference ontology, but rather we show a *possible* way of mapping a given DTD over an ontology representation, and how it is possible to exploit the ontology and the mapping to define quality metrics for an XML document.

3 Defining Quality Metrics Through Reference Ontologies

In this section, we will show how quality metrics for XML documents can be defined with respect to a reference ontology that encodes the domain knowledge required for the cleaning task. The goal of our work is to specify a general set of metrics that can be used to assess the quality of an XML document with respect with different quality dimensions. In particular, we believe that is fundamental to address four dimensions, namely *accuracy*, *completeness*, *consistency* and *currency*. In Section 3.3, we show how quality metrics can be defined for the *completeness* dimension. Other definitions regarding the completeness dimension can be found in [10,7]. These defintions, however, do not deal with XML data. Before presenting our metrics, we need to give some preliminary definitions.

3.1 DTDs and XML Documents

In this section we formally define a simplified model for XML DTDs and documents that will be used throughout the document.

Definition 1. *A* restricted DTD *is a tuple* $D =< T_v, T_c, \tau_r, A, def, attlist, req >$ *where:*

- T_v *is a finite set of value-types;*
- T_c *is a finite set of complex-types;*
- τ_r *is a separate type called the root type;*
- A *is a finite set of attribute types;*
- *for each* $\tau \in T_c \cup \{\tau_r\}$, $def(\tau)$ *is a regular expression called the element type definition of* τ. *The language of the regular expressions used for element type definitions is described by the following grammar:*

$$\alpha ::= \tau_v \mid \tau_c \mid \alpha|\alpha \mid \alpha,\alpha \mid \alpha^* \mid \varepsilon$$

  where ε *denotes the empty content,* $\tau_v \in T_v$,$\tau_c \in T_c$ *and the symbols "|", "," and "*" denote union, concatenation and Kleene closure;*
- *for each* $\tau \in T_c \cup \{\tau_r\}$, $attlist(\tau) \subseteq A$ *is a set of attribute types;*
- *req is a function from* $T_c \times A$ *to* $\{true,false\}$.

Notice that in our simplified model we explicitly disallow mixed content, i.e. elements having both element and text children. Also, value-typed elements, that is an elements containing only one text child, cannot have attributes. The function *req* captures the behavior of DTD specifications wrt the #REQUIRED keyword that can be used as a default value for attributes.

```
<!DOCTYPE Awards [
<!ELEMENT Awards (Movies, BestActorsInLeadingRole)>
<!ELEMENT Movies (Movie*)>
<!ELEMENT Movie (Title, Director?)>
<!ELEMENT Director (Name, Surname)>
<!ELEMENT BestActorsInLeadingRole (Actor*)>
<!ELEMENT Actor (Inmovie*, Name, Surname)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Surname (#PCDATA)>
<!ELEMENT Inmovie (#PCDATA)>
<!ATTLIST Movie Year #PCDATA MovieID #PCDATA #REQUIRED>
]>
```

Fig. 1. The DTD of our running example

Example 1. Figure 1 shows an example of DTD for documents containing information about actors that have been awarded with the Best Actor in Leading Role Academy Award, and movies for which they have been awarded. This DTD can be described as a simplified-DTD following our notation as D=< $T_v, T_c, \tau_r, A, def, attlist, req >$, where:

- $T_v = \{Name, Surname, Title, Inmovie\}$;
- $T_c = \{Movies, BestActorsInLeadingRole, Movie, Actor, Director\}$;
- $\tau_r = Awards$;
- $A = \{Year, MovieID\}$;
- $def(Movies) = Movie*$; $def(BestActorsInLeadingRole) = Actor*$; $def(Movie) = Title, (Director|\varepsilon)$; $def(Actor) = Name, Surname, Inmovie$; $def(Director) = Name, Surname$;
- $attlist(Movie) = \{Year, MovieID\}$;
- $req(Movie, Year) = false$,$req(Movie, MovieID) = true$

We formalize next the concept of XML document valid w.r.t. a given simplified-DTD. In this paper, we will only deal with XML trees that have an associated simplified DTD and are valid with respect to it.

Definition 2. *Given a simplified-DTD* $D =< T_v, T_c, \tau_r, A, def, attlist, req >$, *an XML document valid w.r.t. D is a tree* $< N, L, r, type, value, subel, att >$ *where:*

- N *is a set of non-leaf nodes;*
- L *is a set of leaves and there are two disjoint sets* L_a *and* L_e *such that* $L = L_e \cup L_a$
- $r \nsubseteq N \cup L$ *is a special node called* root.
- *type is a function from* $N \cup L$ *to* $T_c \cup T_a$ *that:*
  - *maps each node* $n \in N$ *into a type in* $T_c \cup \{\tau_r\}$;
  - *maps each node* $l_a \in L_a$ *into a type in* A;
  - *maps each node* $l_e \in L_e$ *into a type in* T_v;
  - *maps the root* r *to* τ_r;

  A node n *is called* element *if* $type(n) \in T_c \cup T_v$, attribute *if* $type(n) \in A$
- *value is function from* L *to string values (including the empty string).*
- *subel is a function from* N *to lists of nodes in* $N \cup L_e$ *such that, if* $n \in N$ *and* $type(n) = \tau$ *and* $subel(n) = [n_1, ..., n_k]$ *then* $type(n_1) \dots type(n_k)$ *is in the regular language generated by* $def(\tau)$;

– *att is a function from* $N \times A$ *to* L_a *such that, for any* $n \in N$ *and* $\tau_a \in A$ *then* $att(v, l)$ *is defined iff* $\tau_a \in attlist(type(n))$*;*

We say in general that there is a parent-child edge from n' to n if $n' \in subel(n)$ or $\exists l \in A, att(n, l) = n'$. The graph defined by the parent-child relation is required to be a tree rooted in r. We will denote the set of children of a node n with children(n).

3.2 Mapping DTDs to Ontologies

In order to define our quality measures, we will assume that a reference ontology has already been defined by a domain expert, and we don't deal here in depth with the problem of formally defining the concept of ontology. Anyway, in order to formalize the notion of mapping of a simplified-DTD to an ontology, we will give here some terminology, together with some minimal assumptions on what the specific ontology language used must allow to express.

Definition 3. *In the following, we will denote a reference ontology with* Σ*. Syntactically,* Σ *is a tuple* $< C, Prop, R >$ *where:*

– *C is a set of* Concepts
– *For each* $c \in C$*,* $Prop(c)$ *denotes a set of named properties. We assume that on properties it is possible to express cardinality constraint that must be satisfied by instances of the concept. In particular, properties may be defined as optional or mandatory.*
– *R is a set of binary relationships of the form* $< c, c' >$ *where c,c' are concepts in C. For these relationships we require that some constraints can be expressed. In particular, given a relationship* $r =< c, c' >$ *in the ontology, we assume the ontology formalism allows:*
  - *to specify cardinality constraints on both concepts c and* c'*;*
  - *to specify a* direction *for the relationship. A relationship on which a direction is defined is said to be a* parent-child relationship*. The concept c is said to have the* role of parent *and the concept* c' *is said to have the* role of child*;*
  - *to specify a constraint over two properties p and* p'*, belonging respectively to* $Prop(c)$ *and* $Prop(c')$*, such that related instances of c and c' will have the same value for p and* p'*. A relationship on which this constraint holds is said to be a* join relationship*. If* $r =< c, c' >$ *is a join relationship with equality constraint over the properties p of c and* p' *of* c'*, we will also write* $r =< c : p, c' : p' >$*.*

Notice that we're considering here only binary relationships, for the sake of simplicity. A generalization to higher arity relationships could be considered.

It is worthwhile to notice that these features of an ontology can be easily mapped to features available in existing formal ontology languages, like OWL or Description Logics.

We define next how a simplified DTD can be mapped to a reference ontology.

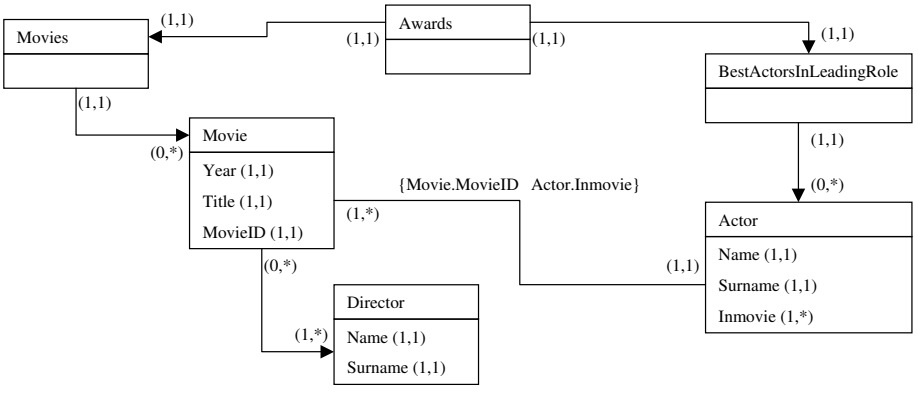


Fig. 2. Reference Ontology

Definition 4 (Mapping). *Let $D =< T_v, T_c, \tau_r, A, def, attlist, req >$ be a simplified DTD and $\Sigma =< D, C, Prop, R >$ an ontology. We define a mapping M between D and Σ as a set of correspondences between types of D and elements of Σ such that:*

- $\forall \tau \in T_c \cup \{\tau_r\}, M(\tau) = c \in C$
- $\forall \tau \in T_c, \forall \tau' \in T_v$ *such that* τ' *appears in* $def(\tau)$*, if* $M(\tau) = c$ *then* $M(\tau, \tau')$ *is a property* $p \in Prop(c)$*;*
- $\forall \tau \in T_c, \forall \tau' \in A$ *such that* $\tau' \in attlist(\tau)$*, if* $M(\tau) = c$ *then* $M(\tau, \tau')$ *is a property* $p \in Prop(c)$*;*

Notice that, given an ontology and a simplified DTD, multiple mappings could in principle be established between them.

Example 2. We have represented graphically in Figure 2 a simple ontology that could be used to describe a conceptual model for the DTD of our example. The mapping between the ontology and the DTD is as follows. Complex types of the DTD are mapped to concepts of the same name in the ontology. Attributes type and value-types in the DTD are mapped on same-name properties of the concepts that have them as subtypes or in the attribute list respectively. The parent-child relationships in the ontology capture the hierarchical structure of the DTD. Notice that this ontology does not simply enforce through cardinality constraints the structural constraints required by the DTD, but also adds other constraints not present in the DTD. In particular, it requires that a join-relationship exists between the Movie and Actor concepts. It also defines as mandatory the property "year" of concept Movie, and adds a minimum cardinality constraint to the parent-child relationship between Movie and Director even if the DTD does not impose these constraints. This choices of course depend on decisions taken by the domain expert that is designing the reference ontology, and may be motivated for example by the fact that the document might, so to speak, "almost" enforce these constraints.

3.3 Defining Quality Measures: Completeness

In the preceding sections we have formalized the concept of simplified DTD and XML document valid wrt a simplified DTD. Furthermore, we have defined how to establish mapping between a simplified DTD and a given ontology. We are now ready to show how it is possible to define quality measures for XML documents based on a reference ontology and the mapping, focusing on a specific quality dimension, namely *completeness*. We will thus introduce a set of completeness- related measures that capture various forms of incompleteness of an XML document.

Definition 5 (value-completeness). *Let n be a node of type $\tau \in T_c$ and $M(\tau) = c$ the corresponding concept in the ontology. Let l be a leaf node of type $\tau' \in T_v$ such that $l \in subel(\tau)$ and $M(\tau, \tau') = p \in Prop(c)$. If p is a mandatory property, the leaf l is said to be* value-complete *if $value(l) \neq \varepsilon$. Notice that leaves corresponding to non-mandatory properties are considered to always be leaf-complete.*

Definition 6 (leaf-completeness). *Let n be a node of type τ and $M(\tau) = c$ the corresponding concept in Σ. Let p be a mandatory property of c. The node n is said to be* leaf-complete *w.r.t. p if it has at least one leaf child l such that $M(\tau, type(l)) = p$. Let $P = \{p1, \ldots, p_n\} \subseteq Prop(c)$ be all the mandatory properties of c. The* degree of leaf-completeness of n, *written $\delta_l(n)$ is defined as the number of properties w.r.t. which n is leaf-complete divided by the cardinality of P.*

Example 3. In Figure 3, the "Inmovie" child of the "Actor" node corresponding to the actor "Russel Crowe" is value incomplete, as it contains the empty string. The "Movie" node corresponding to the movie "American Beauty" is leaf-incomplete w.r.t. the property "year". The degree of leaf-incompleteness of this

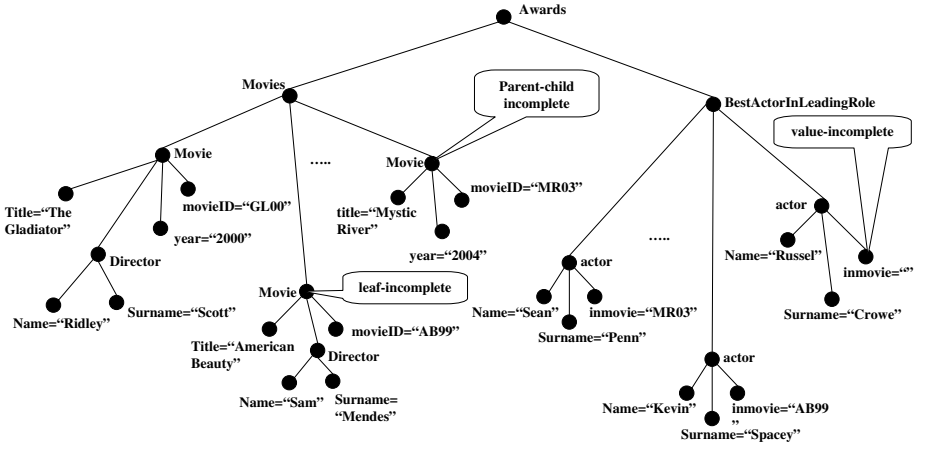


Fig. 3. An XML data tree that conforms to the DTD

node is 0.67 since all three properties of the corresponding concept are mandatory and the node is leaf-complete w.r.t. the other two. Notice that both these problems do not violate the constraints enforced by the DTD. In particular, while the DTD could in principle enforce the requirement that the subnode year is always present, it could not by any means enforce that the value InMovie is present, as the empty string is a valid #PCDATA.

Definition 7 (parent-child completeness). *Let n be a node of type τ and $M(\tau) = c$ the corresponding concept in Σ. Let $r =< c, c' >$ be a directed (parent-child) relationship to which c participates with cardinality $1 \ldots *$ with the role of parent. We say that n is parent-child complete with respect to r iff $\exists$ at least one children of n, n' such that $M(type(n')) = c'$.*

Let now $R_{pc} = \{r_1, \ldots, r_k\}$ be all the parent-child relationships to which c participates with cardinality $1 \ldots *$ Let $C_{pc} = \{cr_1, \ldots, cr_k\}$ be the concepts having role of child in the relationships of R_{pc}. The degree of parent child completeness *of n, written $\delta_{pc}(n)$, is defined as the number of relationships in R_{pc} with respect to which n is parent-child complete, divided by the cardinality of R_{pc}. More formally, lets suppose without loss of generality that $\overline{R_{pc}} = \{r_1 =< c, c_{r_1} >, \ldots, r_s =< c, c_{r_s} >\} \subseteq R_{pc}$ is the set of relationships such that $\forall r_i \in \overline{R_{pc}} \exists n_{r_i} \in subel(n) such\ that M(type(n_{r_i})) = c_i$. Then:*

$$\delta_{pc}(n) = |\overline{R_{pc}}|/|R_{pc}|$$

Example 4. The "Movie" node for movie "Mystic River" shown in Figure 3 is not parent-child complete w.r.t. the relationship between the concept "Movie" and the concept "Director", because it does not have any "Director" subelement. Since the concept "Movie" does not have the role of parent in any other parent-child relationship, the degree of parent-child completeness of the node is 0.

We have defined in a similar way other relationship-based quality metrics, Namely *join-completeness* and *r-completeness*. The definitions of such metrics can be found in [6].

4 Conclusions

We have presented **OXC**, a framework to perform data cleaning over XML documents, given a DTD specification of their schema. **OXC** comprises an architecture for ontology-driven data cleaning based on a methodology that uses an expressive, ontology-based representation of the domain knowledge in order to allow data quality assessment and data cleaning tasks over XML documents. Apart from describing the methodology that comes with **OXC**, the main contribution of this paper is in the formalization of the problem of defining quality metrics for XML documents based on a reference ontology. We have shown here in particular how to define some basic metrics related to the *completeness* quality dimension. Future work will address the definition of metrics related to other fundamental data quality dimensions, namely *accuracy*, *consistency* and *currency*.

References

1. M. Arenas and L. Libkin, *A normal form for XML documents*, ACM Trans. Database Syst. **29** (2004).
2. H. Galhardas, D. Florescu, D. Shasha, and E. Simon, *An Extensible Framework for Data Cleaning*, Proceedings of the 16th International Conference on Data Engineering (ICDE 2000), San Diego, CA, USA, 2000.
3. H.V. Jagadish, L.V.S. Lakshmanan, M. Scannapieco, D. Srivastava, and N. Wiwatwattana, *Colorful XML: One Hierarchy Isn't Enough*, Proceedings of the 2004 ACM SIGMOD Confrence (SIGMOD 2004), Paris, France, 2004,.
4. L. Liu and L. Chi, *Evolutionary Data Quality*, 7th International Conference on Information Quality, Boston, MA 2002.
5. M. Bovee and R.P. Srivastava and B.R. Mak, *A Conceptual Framework and Belief-Function Approach to Assessing Overall Information Quality*, Proceedings of the 6th International Conference on Information Quality, Boston, MA, 2001.
6. Diego Milano, Monica Scannapieco, and Tiziana Catarci, *Using Ontologies for XML Data Cleaning(Extended Version)*, Available on-line at http://www.dis.uniroma1.it/ milano/docs/oxc.pdf.
7. F. Naumann, J.C. Freytag, and U. Leser, *Completeness of integrated information sources*, Information Systems **29** (2004), no. 7.
8. J. Christoph Freytag Rainer Conrad, Dieter Scheffner, *Xml conceptual modeling using uml*, 19th International Conference on Conceptual Modeling, Salt Lake City, Utah, USA, 2000.
9. V. Raman and J.M. Hellerstein, *Potter's Wheel: An Interactive Data Cleaning System*, VLDB, Roma, 2001.
10. M. Scannapieco and C. Batini, *Completeness in the Relational Model: A Comprehensive Framework*, 9th International Conference on Information Quality, Boston, MA, 2004.
11. L.L. Wai, M.L. Lee, and T.W. Ling, *A Knowledge-based Approach for Duplicate Elimination in Data Cleaning*, Information Systems **26** (2001).
12. R.Y. Wang, *A Product Perspective on Total Data Quality Management*, Communications of the ACM **41** (1998), no. 2.
13. M. Weis and F. Naumann, *DogmatiX Tracks down Duplicates in XML*, SIGMOD Conference, Maryland, 2005.

Applying Patterns for Improving Subcontracting Management

Riikka Ahlgren, Jari Penttilä, and Jouni Markkula

University of Jyväskylä,
Information Technology Research Institute,
P.O. Box 35, FIN-40014 University of Jyväskylä, Finland
{ahlgren, penttila, markkula}@titu.jyu.fi

Abstract. This paper studies inter-organizational communication of strategic design information. The focus is on global software subcontracting, where communication problems are common. Software patterns, which have been recognized as a valuable tool in software development, are proposed to be means to facilitate the communication of design information in subcontracting relationship. The position of patterns in subcontracting related processes are studied and the implications of introducing patterns to software subcontracting relationship are analyzed. As a result an evaluation of software patterns' suitability as means for efficient, systematic and explicit communication in managing the subcontracting relationship is presented.

1 Introduction

Rapidly changing technologies and changing market forces are common in contemporary ICT business environment. The ability to respond quickly and efficiently to these changes, referred to as business agility [1], is therefore crucial for enterprises to succeed in this field.

To gain this agility, software business is formed by networks of several players. The players need efficient co-operation to be able to produce the products, applications and services in time [2]. Due to this, subcontracting and subcontracting management has become one of the key success factors for companies to bring the needed flexibility into their operations [3].

As subcontracting stands in the core of business, constant improvement is needed for tools, methods, procedures and management competencies. However, several problems have been encountered in the subcontracting management, particularly in the collaboration of enterprises. The problems in software subcontracting often relate to contracting, requirements engineering, project management, to overall quality of the subcontracted component and foremost to the communication that is needed in the different phases of subcontracting process. [4, 5, 6]

In information and communication technology related system and product development, the most essential content of the communication is design information; information about system, hardware and software, design solutions and their characteristics. The communication needs in subcontracting are identified by Paasivaara [5], and the four communication categories are 1) problem solving, 2) informing and

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 572 – 581, 2005.
© Springer-Verlag Berlin Heidelberg 2005

monitoring, 3) relationship building and 4) decision making and coordination [5]. In this paper the focus is mostly on the second and fourth categories.

Software developer community has adopted software patterns as a means to communicate and document design information [7, 8]. Although pattern origins lay in building architecture, they are successfully applied in software development since mid 90's [8]. Patterns enable reuse of good and verified design solutions in an efficient manner, facilitate communication and are a tool to document the produced code [7]. Patterns' use has gradually increased in software industry, as enterprises begin to see patterns' value as a part of systematic software development.

In this paper, the view of patterns' utilisation is extended from their traditional role as a communication and documentation tool to a design information communication tool that can be used in managing subcontracting relationship. We propose the idea of utilising patterns in a systematic way in subcontracting relationship, study what effects it would have to the related processes and analyse what would be the possible benefits.

This paper is organized in the following way. Software subcontracting, its' processes and communication needs are presented in the second section. In the third section patterns are introduced as a communication facilitator between the subcontracting parties. Patterns' suitability for different subcontracting processes is analyzed in the fourth section. Section five concludes the paper with critical evaluation.

2 Subcontracting

Software companies use subcontracting for adding flexibility to their development operations. Subcontracting decisions are based on the benefits gained through outsourced development processes and tasks. These benefits are either financial or other benefits, including process agility, time-zone-effectiveness, desire to concentrate on core competences, or other specific knowledge advantages achieved by using subcontractors located near the market [3, 9].

In this paper is Capability Maturity Model Integration (CMMI) used as software subcontracting framework. It is widely used framework for process improvement and it consists of best practices that address the development and maintenance of products and services. CMMI integrates several bodies of knowledge, which have been addressed separately in the past. These knowledge areas are for example software engineering, systems engineering and acquisition. [10]

The general processes related to subcontracting can be identified from CMMI Supplier Agreement Management (SAM) process area. The purpose of the process area is to manage the acquisition of products from suppliers, for which there exists a formal agreement. [10]

In CMMI 1.1 SAM [10] the specific goals (SG) and related practices (SP) are:

SG 1 Establish Supplier Agreements:
- SP 1.1-1 Determine Acquisition Type
- SP 1.2-1 Select Suppliers
- SP 1.3-1 Establish Supplier Agreements

SG 2 Satisfy Supplier Agreements:
SP 2.1-1 Review COTS Products
SP 2.2-1 Execute the Supplier Agreement
SP 2.3-1 Accept the Acquired Product
SP 2.4-1 Transition Products

In the CMMI SAM process area, the two specific goals, establishing and satisfying supplier agreements, define the processes required in subcontracting management. In the present context of subcontracting relationship, the specific practices of determining acquisition type (SP 1.1-1) and reviewing COTS products (SP 2.1-1) are not relevant. However, the specific goals and other specific practices present a framework, where the client and supplier communicate, and share and use information.

In the first specific goal, in establishing supplier agreements (SG 1), communication of design information between the client and supplier is needed in sub practices of selecting suppliers (SP 1.2-1) and establishing supplier agreements (SP1.3-1).

In selecting suppliers (SP 1.2-1) the client should select suppliers based on evaluation of their ability to meet the specified requirements and established criteria. The criteria should be established to address factors that are important to the client's project, where the component is integrated. The factors include, for example, supplier's performance records on similar work, engineering capabilities and prior experience in similar applications. Typical work products include, for example, rationale for selection of candidate suppliers, evaluation criteria and solicitation materials and requirements. Evaluation of suppliers' ability to perform the work and risks associated with proposed supplier are also included in the sub-practices of this specific practice. [10]

Trend is to have only few but skilled and close subcontractors, which are more like partners than just subcontractors [4, 11]. This kind of strategic partner management is seen to result in more trusting relationships where power can more and more be given to the subcontractor. Thus, the communication need during the subcontracting process is minimized without the product quality to suffer.

In the sub practice of establishing supplier agreements (SP 1.3-1) the client establishes and maintains a formal agreement with the supplier. The agreement should identify decision making, reporting, requirements and their management, documentation and services and other relevant procedures. Typical work products include statement of work, contracts and licensing agreement. In the specific goal of supplier agreement establishing (SG 1), the included explicit design information plays a crucial role, as it forms the formal base for the following goal of satisfying supplier agreements.

The specific goal of satisfying supplier agreements (SG 2) includes, as specific practices, execution of the supplier agreement, acceptance of acquired product and transition products. Execution of the supplier agreement (SP 2.2-1) means performing activities with the supplier as specified in the agreement. Project management, technical and managerial reviews and other control procedures are typical tasks of the practice.

Acceptance of the acquired product (SP 2.3-1) means ensuring that the supplier agreement is satisfied before accepting the acquired product. Acceptance tasks include definition of the acceptance procedures and their review with all the stakeholders, product verification, documenting of acceptance test results, and definition of an action plan for the products which do not pass their acceptance review or tests.

The goal of the sub practice of transition products (SP 2.4-1) is to ensure smooth transition of the acquired products from the supplier to the project. Planning and evaluating is done to ensure that facilities and training are appropriate and that storage, usage and distribution of the products are done according to specified agreement.

In current research of software development and subcontracting two types of problems have been identified. First are the project-related problems: subcontracting projects are finishing too late and not within budget. Lack of efficient change management practices is a major reason to projects being overtime. Also unrealistic estimation of cost and time or insufficient risk management can cause problems in development projects. Second are the product-related problems: expectations regarding the outcome often do not match with actual results, so that it will be hard to apply the results. Example of this is too complex and difficult requirements to be identified. [4, 12]

These problems are similar to the main problems of in-house software development, but the tools to deal with these problems are not. In a subcontracting setting, the contractor has nearly no influence on the processes of the subcontractor to deal with these problems. The insufficient control in software development implies that subcontracting assessment should pay attention to goal definition and a process structure, establish a trade-off between goals and resources, and provide feedback information. [4]

Information security and protection of intellectual property rights (IPR) is one of the core issues in subcontracting risk management [3]. As subcontracting has become more and more popular in software development, these issues have raised to the limelight of risk management processes. Especially this is important when considering global subcontracting, where valid legislation and its impact can change according to the location of the subcontractor.

In this paper patterns are suggested to be a tool for subcontracting management, for their structure enables them to mediate strategic information to both client and to the subcontractor. Using patterns can help to solve both project and product related problems.

3 Patterns in Subcontracting Relationship

The concept of patterns was brought to software development from building architecture in 1995 [7, 13]. Since then they have been used as a means to collect, store and distribute design information about successful, verified solutions.

Patterns are common, often very well known, solutions to recurring problems. Patterns capture proven solutions in a structured form and build a common vocabulary for communication. They are typically classified to three levels: architectural patterns, design patterns and idioms, and they present the core of the solution in an abstract form, still allowing the implementation to be done in varying ways. Thus, the final result is always unique. [8]

Various roles involved in software development process (e.g. programmers, designers and software architects) can use patterns to improve their understanding and communication, for example about software architecture. Thus, patterns can be used to facilitate communication and also software analysis and description.

Patterns' advantage in communication is their structural composition [14]. This structure reveals pattern consequences and implementation trade-offs, and it helps to keep track of design alternatives, including the ones that were not approved for implementing [14]. This data can be exploited when strategic decisions are made.

Patterns are not necessary in every project phase; instead they can be adopted only in some phases. This is often recommended particularly when introducing patterns in the process for the first time [15]. However, according to [15], best results can be achieved if the patterns are used systematically throughout the process, for architectural view of the application is achieved by ignoring the implementation issues at high-level abstract view of interfacing patterns and traceability and links between high-level view and low-level details are enabled [15]. This traceability can provide the needed early warning signals of the process when something does not go as planned.

Patterns are created by people - inside the enterprise, among the people in the software development network, or they are existing pattern descriptions adopted from external sources. Active generation of new patterns descriptions consume time, as they must be written down, and then shared among right people. This requires systematic processes to support the writing, sharing and using information in a form of divisible pattern descriptions [16].

According to Manns, the adoption and implementation of pattern processes should focus on demonstrating relative advantages and benefits of using patterns, both from the point of individual work and organizational effectiveness [16]. Also adequate level of training and time should be arranged to support individual learning, and patterns and patterns use should be made visible in the organization, including the possible pattern repositories.

To illustrate the possibilities of patterns' use, the following figure 1 describes an exemplary division of labor in subcontracting. The client is responsible of requirement analysis, testing and integration of the developed components. The subcontractor in turn, is responsible for design, coding, unit testing and possible maintenance.

In cases where the client gives the requirements to the subcontractor for design, the set of patterns can either be suggested or dictated by the client, but as well that can be

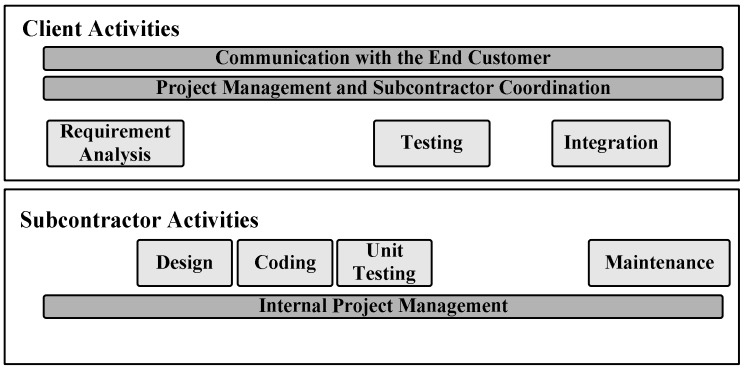


Fig. 1. Exemplary division of labor between client and subcontractor (adopted from [6])

left for the subcontractor to do. Client can then check the planned patterns and accept or make corrections to them. The source, where the patterns are selected from, has significance in the relation. Predefined libraries are probably narrower and more effective to use, but also more risky in business sense, if the patterns are considered as intellectual property of a particular enterprise.

If the application design is done by the client, pattern diagrams and more detailed pattern descriptions can facilitate the subcontractor to add his understanding about the subject matter. Thus, this gives the subcontractor the perspective between analysis and design phases, but reveals only what is essential [15].

In subcontracting relationship, the expectations of client and subcontractor are controversial, for to protect their own business interests, client wishes to deliver as little information as possible. However, the information must be adequate for the subcontractor to fulfill the agreed contract. The subcontractor, in turn, expects the client to inform them as much as possible to ensure the product quality and validity.

From the client point of view, patterns can be used in subcontracting to support subcontracted application development processes. For example sharing and protecting software design information and maintaining control over subcontracted processes can be facilitated by patterns, as is presented in following sections. From the subcontractor's point of view, it is highly important to acknowledge, that the flow of pattern information is bi-directional, from client to subcontractor and vice versa.

As an abstract presentation of design solution, patterns provide a means to deliver information about the system design, without revealing the concrete details of the system architecture, which might be confidential strategic information. The patterns, when used systematically in design process, also provide a means to estimate the required resources and time for developing the component, as well as its properties.

4 Applying Patterns in Subcontracting Management Processes

In previous sections the potential usage of patterns in improving subcontracting management by supporting communication of design information was presented. In the framework of CMMI SAM process area, the possible application of patterns is related to both specific goals of establishing and satisfying supplier agreements (SG 1 and SG 2).

4.1 Patterns in Establishing Supplier Agreements

When selecting suppliers (SP 1.2-1), the client specifies the product or product components to be acquired, based on the overall system architecture where the component is to be integrated. The supplier selection rationale is also specified, which has implications to requirements of design solutions. Furthermore, the potential suppliers are required to present their products and design capabilities, as well as estimate the costs.

Patterns can potentially contribute significantly to supplier selection. They can be used to specify the architecture of the client's system where the component is integrated. They can also be used to specify the core of the design solutions required for the component. In this respect, patterns can be a valuable tool.

Potential suppliers can also present their engineering capabilities and prior experiences in the form of patterns, which they have been using before for similar

applications. Further, client can measure suppliers' skills and use patterns as an evaluation criterion when making selections.

Contracting is an important task of the sub practice of establishing supplier agreements (SP 1.3-1), and some of the tasks in contracting can be facilitated by using patterns, to clarify the client intentions and needs. For example, the requirements for the project, e.g. low cost or fast schedule, can be exemplified by introducing a set of patterns whose consequences imply given outcome. It is important to notice the design trade-offs that the patterns bring with, to avoid incorrect results. Particularly the pattern names and brief pattern summaries can be used to communicate the respected application features to the subcontractor in natural language. Part of acceptance criteria can as well be formulated according to these patterns, which enables comparison between requirements and results.

If client will encourage the pattern processes by the subcontractor, some phases can be written by phase name in the supplier agreement. This provides the client more control over the development process, which is important particularly in early stages of the subcontracting relationship [10]. Control can also be achieved by defining certain processes that should be used during the development process, and some of these processes can be pattern related, for example the processes of pattern acquisition and documenting of used patterns. Though, to require the processes from the subcontractor means, that those processes must be defined and tried out inside the client enterprise as well.

Patterns give the parties a possibility to agree on crucial design solutions in abstract level and also specify possible design alternatives, estimating their cost and effect to the design. Statement of work is a document where the patterns, or group of patterns, expected to be used should be presented. Furthermore, a well known feature of patterns, aid for software documentation, can be utilised in specifying the documentation and transition of maintenance. Based on establishing the supplier agreements, the potential use and advantages of patterns is continuing in satisfying supplier agreements.

4.2 Satisfying Supplier Agreements

In satisfying supplier agreement (SG 2), the specified agreement is satisfied by supplier and client. The component to be produced has to fulfill the requirements of the particular project as well, into which it is later integrated. The design information is used here to verify the component, to follow the progress of the process and the performance of the component, and to test the component and make corrective actions. The documentation and transition of the component to the client for maintenance are as well facilitated by extensive use of design information.

In executing the supplier agreement (SP 2.2-1) patterns provide means to present and communicate design solutions and can aid in documenting and in development progress monitoring. The subcontracted project can be monitored by setting some milestones according to agreed patterns. Thus, the quality of the product and the precedence of the project can be monitored in a concrete way. Patterns can also be used to define the required level of skills of the developers. The skills can be measured and analyzed by the set of patterns that are proved and known by the developers or which should be familiar when starting the subcontracting project.

Intellectual property protection issues are essential in all subcontracting, but particularly in relations, where subcontracting with a specific partner is limited to a temporary project, and not guided by long time investments to specific cooperation contracts. From this viewpoint, use of patterns can be seen as means of protecting ones rights and business secrets for both client and subcontractor. If design information can be distributed systematically in a form of patterns, the client can restrict the need for sharing in-depth information of the implementation environment (e.g. software architecture, other functionalities of the software under construction etc). Subcontractor in turn can use patterns to protect the implementation details of the components to be produced, as long as there is any risk for the deal to be cancelled.

Patterns, when used systematically, can be beneficial in the sub practice of accepting the acquired product (SP 2.3-1), as a means for more efficient acceptance testing. When patterns are used and documented in software design, an experienced programmer can read and interpret the code much quicker and easier than if he or she needs to track the programmers design solutions only from the code.

In transition of the products (SP 2.4-1), the maintenance of the code becomes easier when patterns are used and documented in the design, giving more clear structure to the code. The integration of the subcontracted component with other components is facilitated by the access points of patterns [15].

5 Conclusions

In this paper it has been presented how patterns can be used to facilitate the communication during the subcontracting process and potentially to improve the overall subcontracting management. From this point of view, several areas have been identified, where patterns can have value in total valuation of subcontracting relationship and in managerial activities. This view includes the value of patterns communicated by the client or subcontractor during the subcontracting relationship, both in establishing and satisfying the supplier agreement.

In the SAM process area, in establishing the supplier agreement, the biggest value of patterns lays in the supplier selection, where suppliers can be evaluated and ranked by the patterns they are familiar with. Particularly in contracting the patterns are valuable. They can be used in creating mutual understanding about the requirements and in protecting one's intellectual property rights during the cooperation. when defining the content of shared classified information.

In the special goal of satisfying the supplier agreement the patterns are most valuable in risk and quality management. Patterns possess means to give more explicit guidelines and requirements for the development activities. Specific patterns can be named, which must be used in the development to conform the terms of subcontracting contract. By stating parts of the requirements in the form of patterns, the client company can control the risks of software development, for example required software functionality and how the needed functions are designed and implemented (does the contractor use required patterns). This supports the client particularly in the practices of accepting the acquired product (SP 2.3-1) and product transition (SP 2.4-1). Additionally, the client gets ability to later evaluate the subcontracted software

components, and if necessary, if deviations from contract requirements have been made, claim compensation based on in-adequate or incorrect results.

The evaluation of patterns' use in software subcontracting leads to identify some future research topics, which are essential for identifying both the benefits and requirements to support pattern adoption and implementation processes in software industry. As there are many potential areas of using patterns in software development, systematic use of patterns have many potential advantages. Thus, the requirements and inhibitors of using patterns in subcontracting relations must be identified, for only after that it is possible to evaluate both the overall attractiveness of patterns as a systematic software development tool, as well as the benefits that can be gained in specific software development subcontracting relationship (e.g. patterns in long time strategic software development alliances versus in short time low risk subcontracting).

Essential future research topics include analysis of patterns' economical value in speeding up the development process, and latest standards' suitability for pattern descriptions (for example UML 2). These topics are central in the research area to identify types of projects where patterns are most valuable and also the suitable extent of patterns' use in each type of project.

Acknowledgements

The research work presented in this paper was done in MODPA research project [http://www.titu.jyu.fi/modpa] at the Information Technology Research Institute, University of Jyväskylä. MODPA project was financially supported by the National Technology Agency of Finland (TEKES) and industrial partners Nokia, SysOpen Digia, SESCA Technologies, Tieturi, Metso Paper and Trusteq.

References

1. Gartner UK, The Age of Agility, Gartner Consulting, (2002).
2. Kar, E. v. d., Maitland, C. F., Montalvo, U. W. d.,Bouwman, H., Design guidelines for mobile information and entertainment services: based on the Radio538 ringtunes i-mode service case study, 5th international conference on Electronic commerce, Pittsburgh, Pennsylvania, USA., (2003).
3. Herbsleb, J.,Moitra, D., Global Software Development, IEEE Software, 18 (2001), pp. 16-20.
4. Assmann, D.,Punter, T., Towards partnership in software subcontracting, Computers in Industry, 54 (2004), pp. 137-150.
5. Paasivaara, M., Communication Needs, Practices and Supporting structures in Global Inter-Organizational Software Development Projects, ICSE '03 International Workshop on global Software Development, Portland, Oregon, USA, (2003).
6. Smite, D., A Case Study: Coordination Practices in Global Software Development, Product Focused Software Process Improvement: 6th International Conference, PROFES 2005, Springer, Oulu, Finland, (2005).
7. Gamma, E., Helm, R., Johnson, R.,Vlissides, J., Design Patterns: Elements of reusable object oriented software, Addison-Wesley, Boston, USA, (1995).
8. Buschmann, F., Meunier, R., Rohnert, H., Sommerland, P.,Stal, M., Pattern-oriented Software Architecture. A System of Patterns, John Wiley & Sons, (1996).

9. Heeks, R., Krishna, S., Nicholson, B.,Sahay, S., Synching or Sinking: Global Software Outsourcing Relationships, IEEE Software, 18 (2001), pp. 54-60.
10. Chrissis, M. B., Konrad, M.,Shrum, S., CMMI: Guidelines for Process Integration and Product Improvement, Addison-Wesley, Boston, (2005).
11. Seppänen, V., Evolution of Competence in Software Subcontracting Projects, International Journal of Project Management, 20 (2002), pp. 155-164.
12. Schmidt, R., Lyytinen, K., Keil, M.,Cule, P., Identifying Software Project Risks: An International Delphi Study, Journal of Management Information Systems, 17 (2001).
13. Alexander, C., Ishikawa, S., Silverstein, M., Jakobson, M., Fiksdahl-King, I.,Angel, S., A Pattern Language. (Volume 2). Oxford University press, New York, (1977).
14. Coplien, J., Software Patterns, SIGS Books & Multimedia, New York, (2000).
15. Yacoub, S. M.,Ammar, H. H., Pattern-Oriented Analysis and Design. Composing Patterns to Design Software Systems, Addison-Wesley, Boston, (2004).
16. Manns, M. L., An investigation into factors affecting the adoption and diffusion of software patterns in industry, De Montfort University, United Kingdom, Leicester, (2002).

Evaluation of Strategic Supply Networks

Antonia Albani[1] and Nikolaus Müssigmann[2]

[1] Delft University of Technology,
Chair of Software Engineering,
PO Box 5031, 2600 GA Delft, The Netherlands
a.albani@ewi.tudelft.nl
[2] University of Augsburg,
Chair of Business Informatics and Systems Engineering,
86159 Augsburg, Germany
nikolaus.muessigmann@wiwi.uni-augsburg.de

Abstract. Based on changing market conditions companies are more and more focusing on their core competencies while outsourcing supporting processes to their business partners. The outsourcing of business processes results in a strong dependency between companies and their business partners building so called *value networks*. In order to perform well in such value networks, the *selection* of both direct and indirect suppliers is of main importance. Potential supply networks need therefore to be *identified* and *evaluated* in order to strategically select the appropriate supply network. The selection of adequate supply networks, satisfying evaluation criteria defined by the OEM, is the topic of this paper. It builds on preparatory work done in the area of strategic supply network development, where the identification and dynamic modeling of strategic supply networks has been elaborated, and focuses on the evaluation of strategic supply networks providing the base for supply network selection.

1 Introduction

Driven by drastic changing market conditions companies are facing an increasingly complex competitive landscape. Decisive factors such as globalization of sales- and sourcing-markets, shortened product life cycles, innovative pressure on products, services and processes and customer's request for individual products are forcing companies to undergo a drastic transformation of business processes as well as organizational and managerial structures [1]. The shift from a hierarchical, function-oriented, to a process-oriented organization with a strong customer focus is essential in order to better adapt to fast changing market requirements and to become more flexible while meeting individualized customer demands [2]. Within an enterprise the core business processes need to be identified, improved and (partly-) automated, while at the same time other processes are outsourced to business partners. As a consequence, business processes concerning e.g. product development, market research, sales, production, delivery and services are affected and have to be adjusted and integrated not only within a single company but also over multiple tiers of suppliers. Corporate networks, so called *value networks,* are formed to better fulfill specific customer requests providing customized products on time in the right quality

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 582 – 591, 2005.
© Springer-Verlag Berlin Heidelberg 2005

and for a competitive price [3-6]. In order to perform well in such value networks, the selection, development, management and integration of respective suppliers, located not only in tier-1 but also in the subsequent tiers, is of major relevance for gaining competitive advantages. Modern information and communication technologies – like the Internet, semantic standards, distributed applications, component based, and respectively service-oriented architectures – are necessary in order to sustain the creation and management of dynamic corporate networks [7]. However, at present IT-enabled value networks can be largely found in form of rather small, flexible alliances of professionalized participants. The support of large value networks with multiple tiers of suppliers still causes considerable difficulties. The high degree of complexity resulting from dynamic changes in value networks is the main reason for the lack of practical implementation that is connected with the identification of supply network entities and the modeling of the supply network structure, as well as with the high coordination effort, as described by [8]. Current Enterprise Resource Planning (ERP) systems build the fundamentals for the management and controlling of supply networks but there is a lack of functionality to support dynamic identification, evaluation and qualification of competent partners [9].

Based on previous work done in the area of identification and modeling of value networks [10-12], this paper focuses on the evaluation of potential supply networks in order to support decision making for the network selection in the domain of strategic sourcing. The paper therefore introduces in chapter 2 the domain of *Strategic Supply Network Development (SSND),* which extends the traditional frame of reference in strategic sourcing from a supplier-centric to a supply-network scope. After having identified and modeled potential supply networks they need to be evaluated in order to provide a basis for network selection. The main functionality relevant for evaluating value networks is therefore introduced in chapter 3 and the corresponding evaluation criteria and methods are presented in chapter 4. Conclusions and future work are given in chapter 5.

2 Strategic Supply Network Development

The relevance of the purchasing function in the enterprise has increased steadily over the past two decades. Till the 70ies, purchasing was widely considered an operative task with no apparent influence on long term planning and strategy development [13]. This narrow view was broadened by research that documented the positive influence that a targeted supplier collaboration and qualification could bring to a company's strategic options [14]. In the 80ies, trends such as the growing globalization, the focus on core competencies in the value chain with connected in-sourcing and out-sourcing decisions, as well as new concepts in manufacturing spurred the recognition of the eminent importance of the development and management of supplier relationships for gaining competitive advantages. As a result, purchasing gradually gained a strategic relevance on top of its operative tasks [15].

Based on these developments, purchasing has become a core function in the 90ies. Current empiric research shows a significant correlation between the establishment of a strategic purchasing function and the financial success of an enterprise, independent

from the industry surveyed [16, p. 513]. One of the most important factors in this connection is the buyer-supplier-relationship. At many of the surveyed companies, a close cooperation between buyer and supplier led to process improvements and resulting cost reductions that were shared between buyer and suppliers [16, p. 516].

In practice, supplier development is widely limited to suppliers in tier-1. With respect to the above demonstrated, superior importance of supplier development we postulated the extension of the traditional frame of reference in strategic sourcing from a supplier-centric to a supply-network-scope, i.e., the further development of the strategic supplier development to a strategic supply network development [10-12]. This re-focused the object of reference in the field of strategic sourcing by analyzing supplier networks instead of single suppliers.

In a next step a summary of the functional tasks of the domain of strategic supply network development is given. The tasks within the strategic supply network development can be grouped into 3 main areas as illustrated in Fig. 1: *strategic demand planning*, *strategic supply network modeling* and *strategic supply network qualification* [10].

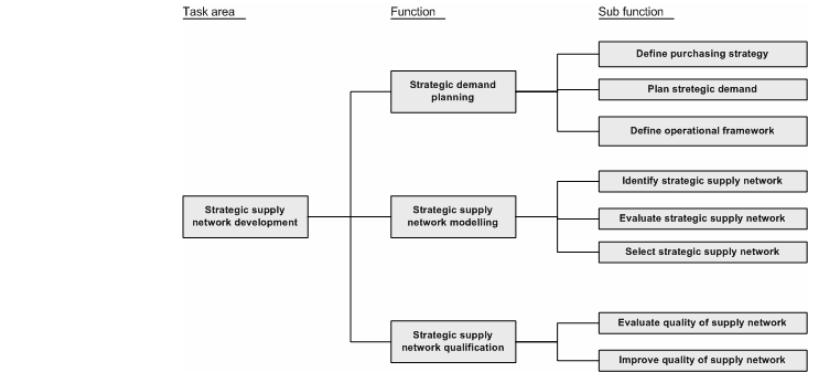


Fig. 1. Functional decomposition diagram for strategic supply network development

Within the function *strategic demand planning,* a corporate framework for all purchasing-related activities is defined. This framework consists of a consistent and corporate-wide valid purchasing strategy (*define purchasing strategy*), a strategic demand planning and demand bundling function (*plan strategic demand*), and the definition of methods and tools to control performance and efficiency of purchasing and to establish a conflict management concept (*define operational framework*).

The function *strategic supply network modeling* provides a methodology for the identification (*identify strategic supply network)*, evaluation (*evaluate strategic supply network*) and selection (*select strategic supply network*) of potential suppliers, not only located in tier-1 but also in the subsequent tiers. Using evaluation criteria such as best cost, shortest delivery time or best quality, and corresponding evaluation methods, the identified supply networks are evaluated. If there is a positive result on

the evaluation, the supply network is selected and contractually linked to the company.

Within the function *strategic supply network qualification,* the quality of a performing supplier network is evaluated using evaluation criteria and evaluation methods (*evaluate quality of supply network*). Dependent on the result of the evaluation, sanctions may be used to improve the quality of the supply network (*improve quality of supply network*).

For the purpose of this paper the focus is set on the modeling of strategic supply networks and specifically on the evaluation sub-task. The reason therefore is that compared to the traditional strategic purchasing the supplier selection process undergoes the most evident changes in the shift to a supply network centric perspective. The expansion of the traditional frame of reference in strategic sourcing requires more information than merely data on existing and potential suppliers in tier-1. Instead, the supply networks connected with those suppliers have to be identified and evaluated, e.g., by comparing alternative supply networks. Since the identification of strategic supply networks builds the basis for the evaluation, we introduce shortly the preparatory work done by the authors in the area of identification and modeling of strategic supply networks [10-12], before explaining in detail the evaluation process in the next chapters.

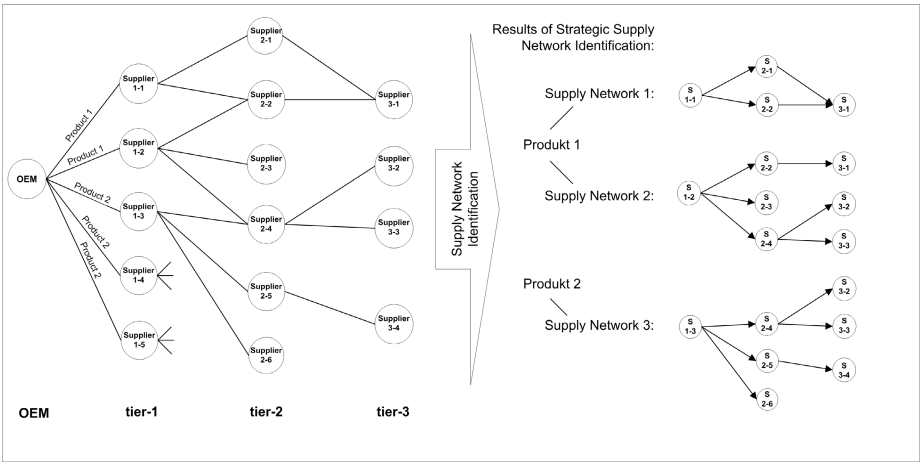


Fig. 2. Example for result of identification process

To model and visualize the network in a structured way, a specific strategic demand for a product to be built is communicated from the OEM to existing and/or potential suppliers. Fig. 2 illustrates an example for an identification process and its results. In the example the OEM is connected to a potential network of suppliers as shown in the left part of Fig. 2. It is assumed that the OEM needs to source two products externally, product 1 and product 2. During the identification process the OEM sends out demands for these products to its strategic suppliers in tier-1. In the example it is assumed that supplier1-1 and supplier1-2 get the demand for product 1

while supplier1-3, supplier1-4 and supplier1-5 receive the demand for product 2. These suppliers check whether they can fulfill the demand internally and if not sent out subsequent demands to their respective suppliers. Each node executes the same process as just described until the demand has reached the last tier. The requested information is then split lot transferred back to the OEM, aggregated and finally visualized as a supply network, in which each participant of the supply network constitutes a network hub. This process may result in the identification of several possible supply networks as shown in the right part of Fig. 2, where e.g. product 1 can be provided by two supply networks, supply network 1 (root node S1-1) and supply network 2 (root node S1-2), whereas product 2 can only be provided by supply network 3. For prove of concept, a prototype tool SSND has been implemented providing the functionality for the identification and dynamic modeling of strategic supply networks [12]. It is now up to the OEM to decide which one of the potential strategic supply networks will be selected to fulfill its original demand. Therefore an evaluation process is necessary which is described in detail in chapter 3.

3 Evaluation of Strategic Supply Networks

Based on identified strategic supply networks, as explained in chapter 2, the basic activities needed for the evaluation are introduced next and summarized in the functional decomposition diagram as shown in Fig. 3.

As part of the *strategic supply network modeling* function the sub function *evaluate strategic supply network* consists of the elementary functions *define evaluation criteria*, *define evaluation method*, *select supply network(s)*, *evaluate supply network(s)* and *visualize evaluation results*.

Evaluation criteria may span from simple facts to highly complex considerations. One of the simplest criteria is the *minimum number of nodes* in the supply network, which can be used to minimize overall complexity of supply networks. Criteria with more complex calculations are e.g. the *shortest total delivery time,* the *minimum total cost* or the *regional only sourcing*. Complex criteria are e.g. *maximize product quality* or *maximize delivery time liability,* since these criteria implicate the evaluation of past experience. While considering critical areas in the supply network it is also of main importance to know which nodes have absolute monopoly with their supply value or which nodes are involved in more than one potential supply network (see S2-2 and S3-1 in the example in Fig. 2).

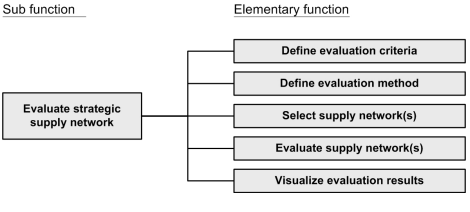


Fig. 3. Functional decomposition of sub function *evaluate strategic supply network*

Evaluation criteria may span from simple facts to highly complex considerations. One of the simplest criteria is the *minimum number of nodes* in the supply network, which can be used to minimize overall complexity of supply networks. Criteria with more complex calculations are e.g. the *shortest total delivery time,* the *minimum total cost* or the *regional only sourcing* (indicating, that only those suppliers are selected, which are located within a certain region). Complex criteria are e.g. *maximize product quality* or *maximize delivery time liability,* since these criteria implicate the evaluation of past experience.

After having specified the evaluation criteria the next step is to determinate the evaluation method within the function *define evaluation method*. In most cases a method is represented by an algorithm which itself is related to the selected evaluation criteria. For example the criteria *minimum number of nodes* involves a simple algorithm counting the nodes in potential tree graphs, comparing the results and selecting the tree (supply network) with the least number of nodes. Assuming that the identification process will provide several possible supply networks for different products the function *select supply network(s)* will select the supply networks related to a specific product in order to *evaluate* them using the evaluation criteria and methods as just introduced above. The result of the evaluation process is a rated list of all supply networks related to a specific product. This list forms the basis for defining the supply networks which can be selected in order to produce the specific product.

4 Evaluation Methods and Criteria

Strategic purchasing within a company is obliged to reduce cost and to increase quality and efficiency within the purchasing business function. Obviously this applies also while working with strategic supply networks. Business objectives will affect the evaluation process. From these objectives and the related requirements evaluation criteria will be deduced which lead to evaluation methods and algorithms respectively. Table 1 shows examples of business objectives and assigns evaluation criteria and methods to them.

In order to automate the evaluation process it is important to define algorithms as basis for a software implementation. Supply networks can be treated as directed routed trees (out-tree for demand process, in-tree for fulfillment process) [17, 18] consisting of nodes representing companies and edges representing flow of information and goods as shown in Fig. 4.

Table 1. Examples for business objectives and related evaluation criteria and methods

Business objective	Requirement	Criterion	Method
"To work with small and concise supply networks"	Minimize number of nodes in supply network	Number of nodes	- count number of nodes - identify minimum - select supply network
"Reduce cost of purchasing"	Minimize cost of sourcing external goods	Cost of internal sourcing Cost of external sourcing	- Calculate sum of sourcing per node - Summate cost for all nodes up to the root node (tier-1 node)
"Reduce procurement lead time"	Minimize procurement lead time	Delivery time of node Transport time between nodes	- Identify paths within supply network - Calculate overall delivery and transport time per path - Identify maximum per supply network - Select supply network with minimum of total delivery time

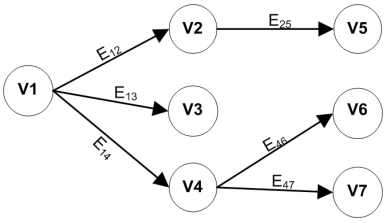


Fig. 4. Example of a supply network as a directed out-tree

A graph G is represented by a set of nodes (V) and a set of edges (E)

$$G = (V,E) \tag{4.0}$$

In the example illustrated in Fig. 4 the set of nodes is:

$$V = \{V1,V2,V3,V4,V5,V6,V7\} \tag{4.1}$$

The set of edges is represented by

$$E = \{E_{12},E_{13},E_{14},E_{25},E_{46},E_{47}) \text{ or} \tag{4.2a}$$

$$E = \{(V1,V2),(V1,V3),(V1,V4),(V2,V5),(V4,V6),(V4,V7)\} \tag{4.2b}$$

As shown in Table 1 there will be multiple evaluation criteria per node and per edge. Therefore it is suggested to introduce an evaluation vector for nodes and edges as illustrated in Fig. 5. The elements of this vector will be the criteria e.g. *cost of purchasing* or *procurement lead time*, but also indicators like *product quality* or p*rocurement lead time quality*. ec_{Vi} is the evaluation vector for a node and ec_{Eij} the evaluation vector for an edge. Example evaluation criteria are c_x related to cost of purchasing and t_x related to procurement lead time.

$$ec_{V_i} = \begin{pmatrix} c_i \\ t_i \\ \dots \end{pmatrix} \qquad ec_{E_{ij}} = \begin{pmatrix} c_{ij} \\ t_{ij} \\ \dots \end{pmatrix} \tag{4.3a,b}$$

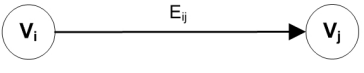


Fig. 5. Evaluation vector for nodes and edges

This paper will concentrate on the description of three example criteria – *number of nodes*, *cost of purchasing* and *procurement lead time* – and will describe the related methods (algorithms, calculation schema) and will suggest a ranking process in order to select supply networks based on a combination of criteria.

Number of nodes

The evaluation of a supply network (G) using the number of nodes (n(G)) criterion is implemented by the following algorithm:

$$n(G) = | V(G) | \tag{4.4}$$

This algorithm simply counts the number of nodes for the supply network.

Cost of purchasing

While calculating the cost of purchasing (c(G)) it is assumed that there is a cost element related to a node (cnVx) (e.g. production cost), a cost element which is related to an edge (ceEx) (e.g. transportation cost) and a total cost element (cVx). Starting from the leaves (end nodes) of the tree a repeated backward calculation is conducted summing up the cost elements up to the root node. As for the example in Fig. 4 the calculation scheme will be:

$$c_{V6} = cn_{V6};\ c_{V7} = cn_{V7};\ c_{V5} = cn_{V5} \tag{4.5a}$$
$$c_{V4} = cn_{V4} + ce_{E46} + c_{V6} + c_{E47} + c_{V7} \tag{4.5b}$$
$$c_{V3} = cn_{V3} \tag{4.5c}$$
$$c_{V2} = cn_{V2} + ce_{E25} + c_{V5} \tag{4.5d}$$
$$c = c_{V1} = cn_{V1} + ce_{E12} + c_{V2} + ce_{E13} + c_{V3} + ce_{E14} + c_{V4} \tag{4.5e}$$

Transforming this example into a generic calculation scheme the following formula results:

$$c(G) = \sum (cn_j + ce_k) \quad (\forall j,k : j \in V, k \in E) \tag{4.6}$$

Procurement lead time

The procurement lead time consists of time elements of nodes (tn_{Vx}) (e.g. production time or delivery time) and time elements of edges (te_{Ex}) (e.g. transport time). In order to calculate the maximum procurement lead time for the complete supply network each node (which is not an end node) has to calculate the maximum procurement lead time of all out-tree paths. As for the example in Fig. 4 the calculation scheme will be:

$$t_{V6} = tn_{V6};\ t_{V7} = tn_{V7};\ t_{V5} = tn_{V5} \tag{4.7a}$$
$$t_{V4} = tn_{V4} + max\{(te_{E46} + t_{V6}),(te_{E47} + t_{V7})\} \tag{4.7b}$$
$$t_{V3} = tn_{V3} \tag{4.7c}$$
$$t_{V2} = tn_{V2} + te_{E25} + t_{V5} \tag{4.7d}$$
$$t = t_{V1} = tn_{V1} + max\{(te_{E12} + t_{V2}),(te_{E13} + t_{V3}),(te_{E14} + t_{V4})\} \tag{4.7e}$$

A generic calculation scheme is suggested as follows: Being NG(k) all adjacent neighbors of node k the procurement lead time (tk) of node k is the sum of the internal node lead time (tnk) and the maximum of the procurement lead times of all adjacent neighbor nodes (being Ekj the edges connecting node k with its adjacent neighbor nodes):

$$t_k(G) = tn_k + \max\left\{(te_{Ekj} + t_j) : j \in N_G(k)\right\} \tag{4.8}$$

This calculation scheme can be repeated for each node in the tree. The overall procurement lead time of the supply network can be calculated setting k = 1 (being 1 the root node of out-tree G).

Combinations of criteria

Using the methods introduced above it is possible to form a total evaluation vector (e_i) for each supply network (G_i):

$$e_i = \begin{pmatrix} ec_i \\ et_i \\ en_i \end{pmatrix} \tag{4.9a}$$

with

$$ec_i = c(G_i);\ \ et_i = t_1(G_i);\ en_i = n(G_i) \tag{4.9b}$$

The identification process may find m valid supply networks (G(i); i = {1,..,m}) all able to fulfill the original demand. Therefore the evaluation process needs to compare m evaluation vectors in order to select the supply network, which meets the evaluation business objective best. The evaluation business objective may be based on a single criterion (e.g. "min. number of nodes") but could also be a combination of criteria (e.g. "min. cost of purchasing and min. procurement lead time"). The combination of criteria can be handled by introducing a weight vector:

$$w = \begin{pmatrix} wc \\ wt \\ wn \end{pmatrix} \tag{4.10}$$

This allows the OEM to prioritize criteria, e.g. w={0,0,1} means "min. number of nodes" or w={0.5, 0.5, 0} means "min. cost of purchasing and min. procurement lead time". The evaluation vectors are therefore multiplied with the weight vector resulting in a ranking list (ri(G)) for the m supply networks:

$$r_i(G) = (wc * ec_i) + (wt * et_i) + (wn * en_i) \tag{4.11}$$

The ranking list is then sorted in ascending order having the supply network at position 1, which meets the evaluation business objective best. It is now up to the OEM to decide, whether the top ranked supply network is selected for purchasing or whether, as in real world scenarios often done, the demand is split and purchasing is done with e.g. the top two ranked supply networks.

5 Conclusion and Future Work

Due to drastic changes in the domain of strategic purchasing, from a supplier-centric to a supplier-network perspective, the selection of adequate suppliers is of main importance for forming value networks. In order to select the respective supply networks they need to be identified and evaluated. Based on preparatory work done in the area of identification and modeling of strategic supply networks, this paper provides algorithms based on vectors containing different evaluation criteria reflecting the relevant business objectives. Because of different possible combinations of criteria weights are introduced for each evaluation criteria in order to provide a ranking list necessary for the selection of the best supply networks. The evaluation vector introduced in this paper is based on elementary criteria and needs to be extended in the future with more complex business objectives. Additionally the weight vector needs to be refined in order to meet advanced business requirements.

References

1. Burtler, P., et al., A Revolution in Interaction. The McKinsey Quarterly, 1997. 1/97: p. 4-23.
2. Österle, H., Business Engineering - Prozeß- und Systementwicklung: Entwurfstechniken. 2 ed. Vol. 1. 1995, Berlin: Springer.
3. Malone, T.W. and R.J. Lautbacher, The Dawn of the E-Lance Economy. Harvard Business Review, 1998(September-October): p. 145 - 152.
4. Warnecke, H.-J., Vom Fraktal zum Produktionsnetzwerk. Unternehmenskooperationen erfolgreich gestalten, ed. J.H. Braun. 1999, Berlin.
5. Pine, B.J., B. Victor, and A.C. Boynton, Making Mass Customization Work. Havard Business Review, 1993. 36(5): p. 108-119.
6. Tapscott, D., D. Ticoll, and A. Lowy, Digital Capital: Harnessing the Power of Business Webs. 2000, Boston.
7. Kopanaki, E., et al. The Impact of Inter-organizational Information Systems on the Flexibility of Organizations. in Proceedings of the Sixth Americas Conference on Information Systems (AMCIS). 2000. Long Beach, CA.
8. Lambert, D.M. and M.C. Cooper, Issues in Supply Chain Management. Industrial Marketing Management, 2000. 29(1): p. 65-83.
9. Angeli, R. Aufbau und Koordination dynamischer Unternehmensnetzwerke. in Wissenschaftssymposium Logistik der BVL. 2002: Huss Verlag.
10. Albani, A., et al. Component Framework for Strategic Supply Network Development. in 8th East-European Conference on Advances in Databases and Information Systems (ADBIS-04), LNCS 3255. 2004. Budapest, Hungary: Springer Verlag.
11. Albani, A., et al. Dynamic Modelling of Strategic Supply Chains. in E-Commerce and Web Technologies: 4th International Conference, EC-Web 2003 Prague, Czech Republic, September 2003, LNCS 2738. 2003. Prague, Czech Republic: Springer-Verlag.
12. Albani, A., C. Winnewisser, and K. Turowski. Dynamic Modelling of Demand Driven Value Networks. in On The Move to Meaningful Internet Systems and Ubiquitous Computing 2004: CoopIS, DOA and ODBASE, LNCS. 2004. Larnaca, Cyprus: Springer Verlag.
13. McIvor, R., P. Humphreys, and E. McAleer, The Evolution of the Purchasing Function. Journal of Strategic Change, 1997. Vol. 6(3): p. 165 - 179.
14. Ammer, D., Materials Management. 2. ed. 1968, Homewood.
15. Kaufmann, L., Purchasing and Supply Management - A Conceptual Framework, in Handbuch Industrielles Beschaffungsmanagement. 2002, Hahn, D, Kaufmann, L. (Hrsg.): Wiesbaden. p. 3 - 33.
16. Carr, A.S. and J.N. Pearson, Strategically managed buyer - supplier relationships and performance outcomes. Journal of Operations Management, 1999. 17: p. 497 - 519.
17. Jungnickel, D., Graphs, networks and algorithms. Vol. 2. 2005, Berlin: Springer.
18. Ahuja, R., T. Magnanti, and J.B. Orlin, Network flows: Theory, algorithms and applications. 1993, New Jersey: Prentice-Hall.

Self Modelling Knowledge Networks

Volker Derballa[1] and Antonia Albani[2]

[1] Augsburg University, Chair of Business Informatics and Systems Engineering,
86135 Augsburg, Germany
volker.derballa@wiwi.uni-augsburg.de
[2] Delft University of Technology, Software Technology,
Mekelweg 4, 2628 CD Delft, The Netherlands
a.albani.@ewi.tudelft.nl

Abstract. What the scope of knowledge management (KM) is concerned, the focus of attention is shifting towards inter-organisational aspects resulting in new requirements for the KM process. This paper introduces the concept of self modelling knowledge networks supporting KM in networks by means of dynamic self-configuration. Apart from introducing the concept, technical issues and design aspects of the implementation are discussed and a component model for an inter-organisational knowledge management system is introduced.

1 Introduction

The imperative for managing knowledge is stressed by a wide array of recent publications ranging from information science to strategic management, substantiating their proposition with the tremendous changes in the context organisations are operating today. Taking into account the complexity of certain value creating processes, the necessary knowledge is often not available within one organisation, but has to be merged and combined from several sources: Partnerships, alliances, consultants, suppliers. For that reason, knowledge management literature and research projects are increasingly shifting their attention from intra-organisational to inter-organisational aspects (e.g. [4, 10, 14, 16]). However, the question of how inter-organisational knowledge management can be realised, is up to now not sufficiently answered [8]. In the context of knowledge networks, that includes the identification of scattered knowledge assets, the visualisation and modelling of the network structure as well as the operation of the whole knowledge network.

This paper contributes to this area of research by introducing the idea of self modelling knowledge networks. In section 2, we describe the resulting concept and present the relevant tasks and information objects related to the domain of knowledge management in a network perspective. In order to automate dynamic modelling, a component model has been derived, building the basis for further implementation of such a system. The component model is introduced in section 3. Conclusions and future work are given in section 4.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 592 – 601, 2005.
© Springer-Verlag Berlin Heidelberg 2005

2 Self Modelling Knowledge Networks

Self modelling knowledge networks can provide a mechanism that enables flexible knowledge retrieval and management across several nodes in a network of firms or independent organisational modules and thus support inter-organisational KM. It is a decentralised system without central servers or directories. Depending on its particular role, every node has the capability to save data, receive and send demands. With client-server functionality, direct communication between the individual nodes and autonomy of the nodes, typical characteristics of centralised P2P-Networks are present. For the area of KM, that is considered a very promising approach [18], as there is typically no common infrastructure in dynamic value networks to support KM activities.

At the core of the concept of self modelling knowledge networks is the notion that the network nodes can be identified by applying the pull principle. With the pull principle, a network node at the beginning of a (sub-) network can identify potential nodes, i.e. knowledge suppliers, in a subsequent tier by performing a knowledge demand specification using standardised ontologies. The mapping of the different ontologies can be achieved through ontology mapping methods as described in [7, 9]. With this information, primary requirements and dependent requirements can be identified and the respective information can be communicated, sending a demand to the network nodes, i.e. the potential suppliers of knowledge. This procedure is repeated by the nodes in the respective tiers until the final tier is reached. Then, the information from the nodes further upstream is aggregated and split-lot transferred to the initiating node. Every node in tier-x receives demands from clients in tier-(x-1) and communicates sub-demands, depending on the demand received, to the relevant knowledge suppliers in tier-(x+1). Since every node repeats the same procedure, a knowledge seeker receives back aggregated information from the whole, dynamically built network based on a specific demand sent at a specific time. Having the fact that the seeker-supplier relationship in value networks may change over time, new dynamically modelled networks – which may differ from the actual ones – are built whenever sending out new demands to the suppliers in the subsequent tiers. The following example demonstrates the idea:

A large manufacturer of diesel engines for power generation plans to adapt the marketing and sales strategy for a particular engine type in order to focus on a specific industry (e.g. mining industry). For that purpose, general information (i.e. contact details of mining companies etc.) as well as specific knowledge concerning the operating experiences and conditions in this industry context is required. That comprises explicit knowledge manifested in service reports, warranty claims, performance data sheets, etc. and implicit knowledge of sales managers, service engineers, technicians, claims managers. This knowledge is widely scattered across several nodes in a network-like structure: the manufacturer, autonomous sales offices, foreign representative offices, engineering companies, consultants, authorised repair shops, and finally the end user, i.e. the mining company. As each of those participants was or is at some stage involved in sales, service, after-sales, warranty claims or operating activities, they all can be considered potential sources for the knowledge required. Lacking a central IT-infrastructure that comprises all participants, the

process of retrieving knowledge cannot be conducted without considerable manual and time-consuming effort.

The concept of self modelling knowledge networks is illustrated in the following as shown in Fig. 1. The figure on the left shows a complete demand driven network composed of existing (highlighted nodes) and alternative sub-networks. Existing sub-networks are those the knowledge seeker already uses. Alternative sub-networks are networks, which are built by sending a demand for a specific knowledge artefact to new chosen knowledge suppliers, with yet no direct relation to the knowledge seeker. The whole network is demand driven, implying that the knowledge network may be different for every demand and for every knowledge seeker. The knowledge seeker communicates a specific knowledge demand to existing and selected alternative nodes in tier-1. Subsequently, the nodes in tier-1 report the corresponding sub-demands to their respective suppliers. Having aggregated the information of all nodes, the node 1-2 adds its own information before split-lot transferring it to the knowledge seeker. Fig. 1 on the right highlights an alternative knowledge sub-network fulfilling the requirements regarding a specific knowledge demand.

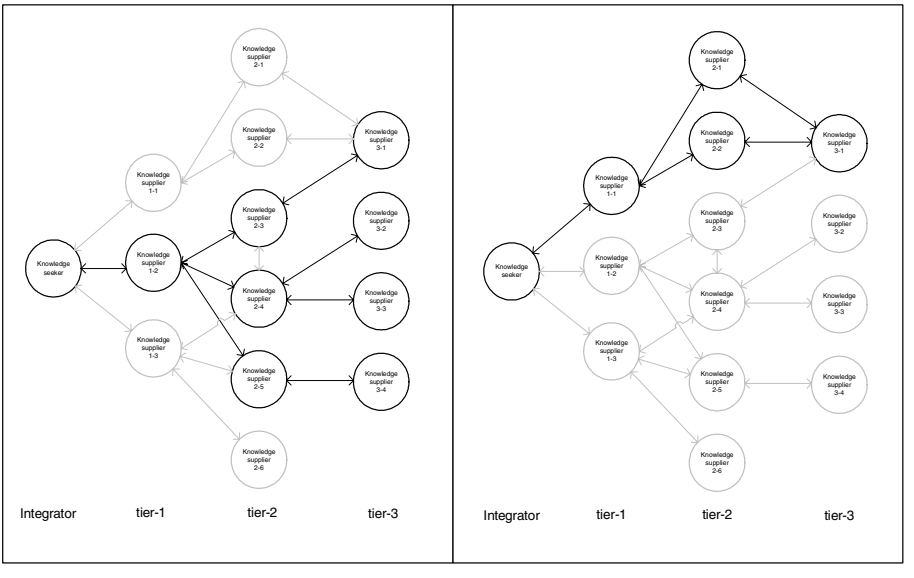


Fig. 1. Left: Knowledge network Right: Alternative knowledge network

2.1 Description of Functional Tasks for the Domain of Self Modelling Knowledge Networks

The functional tasks of the domain of self modelling knowledge networks are defined next. Those tasks are derived from main KM activities as described for example in [13]. In this context, we will only consider the tasks that can be automated by a business application. The functional tasks have been illustrated in a function decomposition diagram (c.f. Fig. 2).

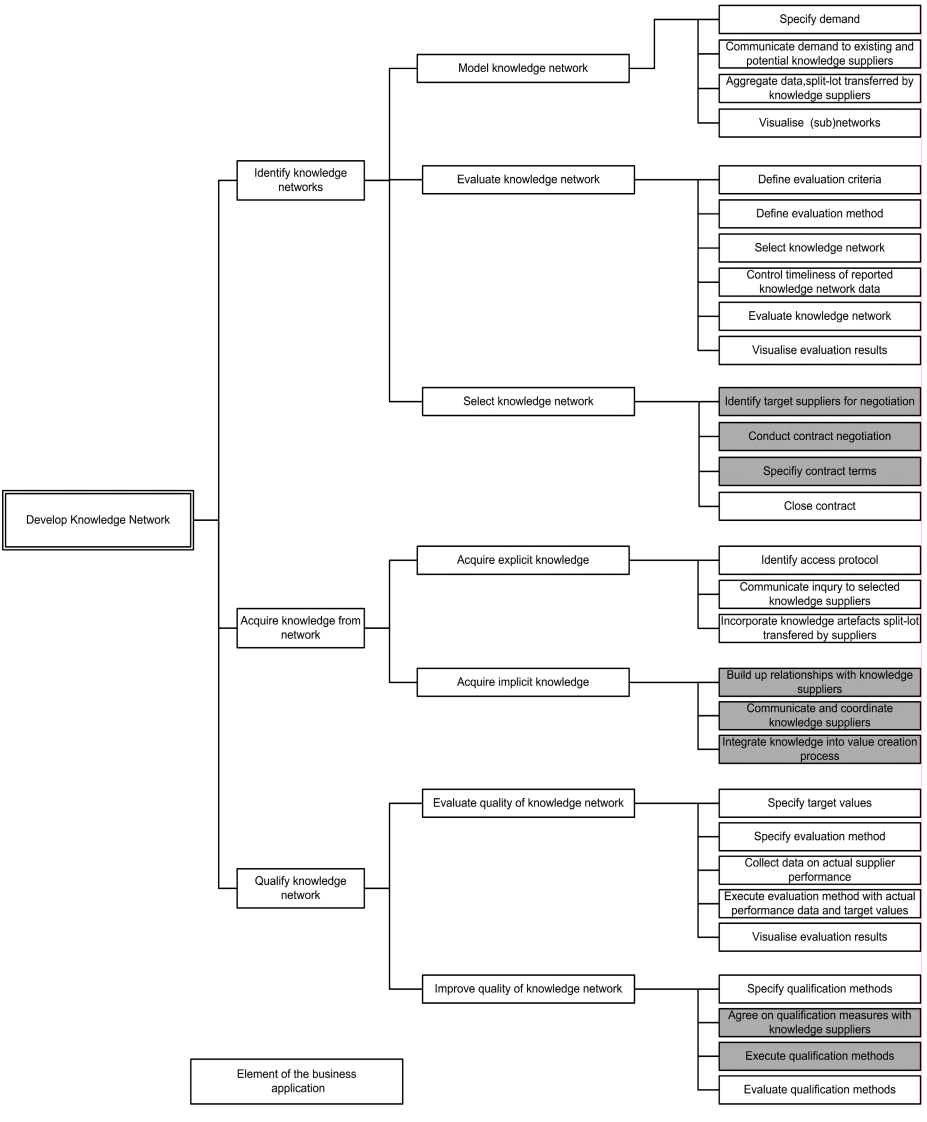


Fig. 2. Functional decomposition for the domain of self modelling knowledge networks

Task "*Identify knowledge network*": This task comprises the process of detecting knowledge and undergoes the most evident changes in the shift to a network perspective. In the conventional process, transparency about available knowledge is created by an internal and external screening process of knowledge sources, i.e. knowledge embedded in human actors or in knowledge artefacts. Based on a network perspective, knowledge identification requires more information than merely data on existing and potential nodes in tier-1. Instead, the knowledge networks connected with those suppliers have to be identified and evaluated (e.g. by comparing alternative

knowledge networks). Therefore, the task of selecting knowledge suppliers is part of the process that leads to the modelling of knowledge networks. The perception of the network as dynamic network constitutes the basis for the identification of knowledge networks.

Task "*Acquire knowledge from network*": This task comprises the process of exchanging and distributing knowledge. Again, the focus shifts from exchanging and distributing knowledge in one single organisation – in case the knowledge is already available there – or from the a single external source – in case the knowledge is acquired externally – to the network perspective. Here, knowledge or knowledge artefacts are distributed across several nodes, with each node adding its partial contribution to the fulfilment of the knowledge demand. After having identified the relevant knowledge artefacts that have to be retrieved, knowledge acquisition describes to process of integrating the external knowledge into the value creation process. In this context, the mode of knowledge – tacit or explicit – is distinguished. The integration of tacit knowledge – embedded in a human actor – cannot be automated. Thus, the capacity of the business application ends here. In that case, the business application serves as an initiation tool. The integration of explicit knowledge (e.g. knowledge stored in databases) can be automated. For that purpose, the access protocol has to be negotiated to allow the knowledge seeker's information systems to access the supplier's data. After that, the knowledge demand is communicated, i.e. an inquiry is conducted. The relevant knowledge artefacts are then split-lot transferred to the knowledge seeker.

Task "*Qualify knowledge network*": This task includes some elements of the KM controlling process, which in the network context encompasses the evaluation of the knowledge nodes regarding their contribution towards the knowledge goals of the knowledge seeking organisation. In addition to the selection of suitable knowledge networks, the performance improvement of strategically important networks is one of the major goals. Main prerequisite is the constant evaluation of the actual performance of selected knowledge networks by defined benchmarks. The application should support respective evaluation methods enabling the user to identify imminent problems in the network and to initiate appropriate measures for the qualification of network partners.

Having identified and characterised the tasks for the domain of self modelling knowledge networks, it is necessary to determine the relevant information objects involved in the modelling of the networks.

2.2 Description of Information Objects for the Domain of Self Modelling Knowledge Networks

The resulting data model is listed as UML class diagram [12] in Fig. 3 and is described next: A whole *knowledge network* specified by a knowledge demand is a network of *knowledge suppliers* with the potential of providing knowledge to a *knowledge seeker*. With the affiliation of *knowledge suppliers* to a *knowledge network* and with the identification of predecessors and successors of the *knowledge suppliers*, the whole network is defined. At a particular time, each node provides information about *financial data, track record, knowledge range* (e.g. competences) or more. This

information is known as *supplier-generated data* and is used to evaluate potential *knowledge suppliers*.

In addition, the *knowledge seeker* generates own data, called *knowledge seeker generated data*, specifying the performance of individual *knowledge suppliers* when cooperating with them. Examples for data generated by the *knowledge seeker* are *target performance data* and *actual performance data*. *Target performance data* act as guidelines for the supplier, whereas the *actual performance data* relate to the work performed, as measured by the *knowledge seeker*. The *knowledge seeker* holds with the acquisition of *supplier-generated data,* the definition and the measurement of performance data for different knowledge networks all the information needed to evaluate the knowledge networks. Different *evaluation methods* therefore are defined by different *evaluation criteria*.

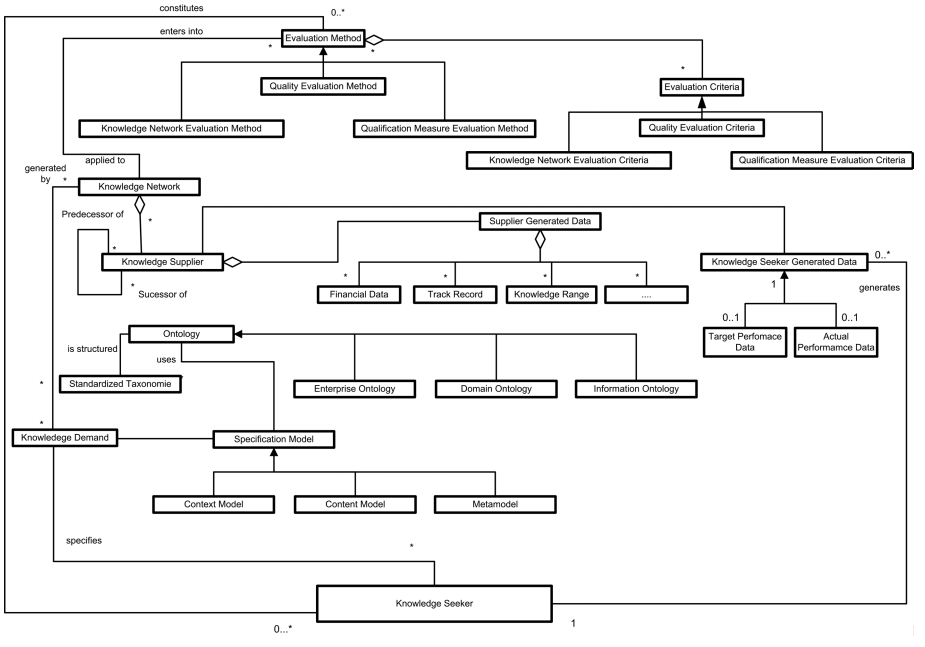


Fig. 3. Data model for the domain of self modelling knowledge networks

A *knowledge seeker,* issuing *a knowledge demand,* specifies his demand by using a *specification model*. The *specification model* comprises *context model, content model and meta model* (c.f. [1]). The *context model* uses *enterprise ontology* to determine the possible creation context of the knowledge artefact. The *content model* uses *domain ontology* containing the relevant concepts, whereas the *metamodel* uses *information ontology* comprising all non-content and non-context specific concepts (e.g. author/owner, location, availability etc.).

The data model and the functional-decomposition diagram derived from the domain analysis are the basis for the development of a system supporting companies

in identifying and developing their specific knowledge networks in order to facilitate flexible adaptability to the changing knowledge requirements. At this stage, we derived a component-based model as basis for implementation. The component model is introduced in the next section.

3 Component Model

To provide a basis for automated modelling of demand driven knowledge networks, a component model for the domain of self modelling networks has been developed, offering basic functionality for the modelling of and collaboration in knowledge networks. The component model is based on the business component technology as defined in [19]. The principle of modular black-box design has been chosen, allowing different configurations of the final system – by combining different components regarding the need of the specific node – ranging from a very simple configuration to a very complex and integrated solution. The system therefore will not only provide basic functionality needed on each network node in order to participate in the network, but it will also provide the possibility of adding new functionality (e.g. adding a component for the evaluation of networks).

Due to space restrictions only one single sub-phase of the process is described next, namely the identification of business components based on the functional-decomposition diagram and on the semantic data model introduced in the previous sections. With the Business Component Identification (BCI) method (c.f. [2]), relationships between business tasks and information objects are defined and grouped. In Fig. 4 on the left, the relationships for the domain of self modelling knowledge networks are shown. The business tasks are gained from the functional-decomposition diagram (c.f. Fig. 2) and are listed left on the table. Relevant groups of data are gained from the data model (c.f. Fig. 3) and utilised for the BCI method as information objects, listed on top. An example relationship would be the creation "c" of the knowledge demand object when specifying the demand; "u" is used for use. Three areas result for the domain of self modelling knowledge networks in changing the order of tasks and information objects in the matrix. The three areas are potential business components. The first business component offers services for the specification and administration of the demand and is called *knowledge demand administration* component (c.f. Fig. 5 on the left).

The second business component is responsible for the development and visualisation of the networks – *knowledge network development* – in aggregating and managing the data received and in providing visualisation services. The *evaluation* component provides services in the area of evaluation methods and criteria. The dependencies between the single components are visualised by arrows.

Having identified the single business components for the domain described, a component model has been designed including additional components implementing non-functional tasks.

The component model is shown in the middle of Fig. 5 in accordance with the notation of the Unified Modelling Language [12]. Additionally to the business components introduced above, the component model provides two system components – *persistence manager* and *collaboration manager* – responsible for the

technical administration of the data and for the collaboration between network nodes. The information managed by the single business components is made persistent through the *persistence manager*. The main reason for introducing the persistence manager is based on the idea of having business components concentrating on the business logic while having system components taking care of implementation specific details. This has an impact on the distribution of the system on network nodes, having the fact that different companies use different physical database systems as data storage. The component model handles that situation by letting the persistent manager take care of implementation specific details without affecting the business logic of the system.

The system provides two semantic storages for data. The *network database* stores all networks containing the aggregated information of suppliers contributing to a specific demand. For each demand, a new network is generated by split-lot transferring data from all suppliers and aggregating the information in the network development component. Such a network is then stored in the network database through the services provided by the persistent manager and called by the knowledge network development component. This information can be retrieved in order to visualise and develop the networks.

The *performance database* provides storage for the companies' performance information. Example clients requesting collaboration services from the system can

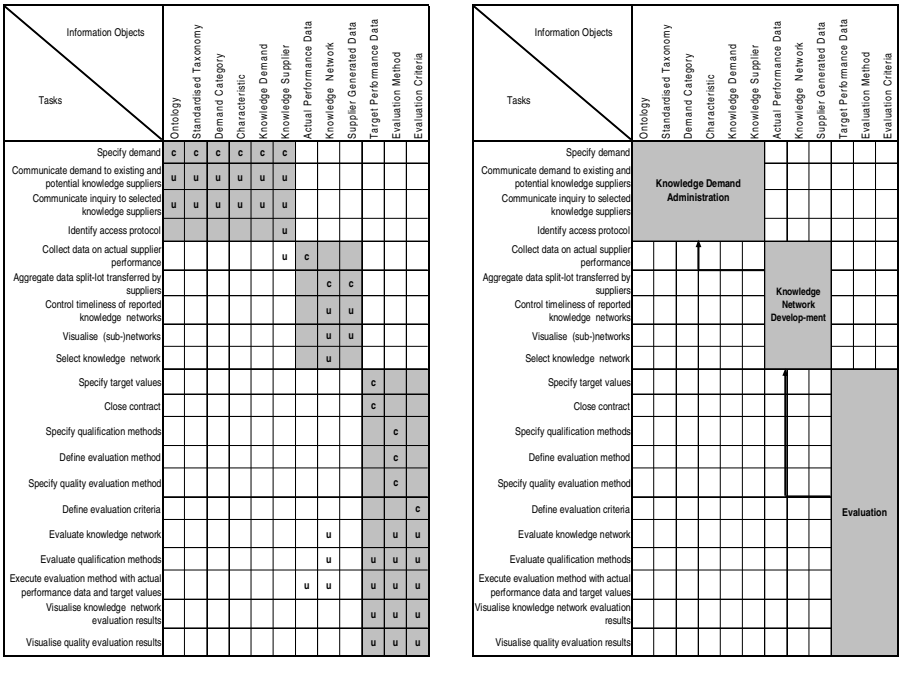


Fig. 4. Business component identification (BCI) method

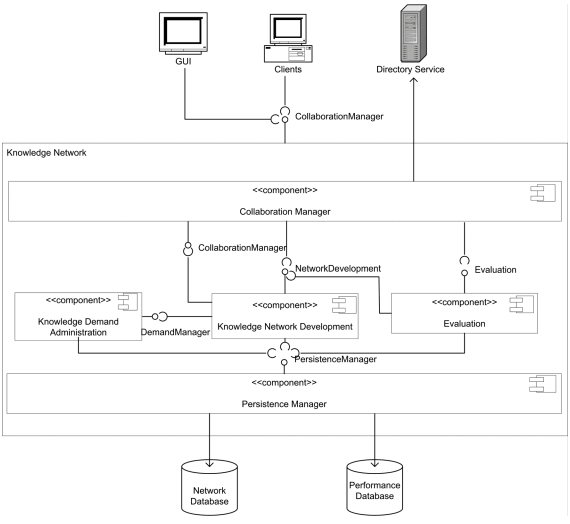


Fig. 5. Component model for the domain of self modelling knowledge networks

either be graphical user interfaces (GUI), asking for data, e.g. to visualise strategic networks, or other network nodes sending demands to suppliers. The collaboration in the system is executed by the *collaboration component*. Regarding the complexity of collaboration in inter-enterprise systems including technical details (e.g. communication protocols) we refer to previous work (e.g. [3, 6]) for further information.

4 Conclusion

This paper substantiates the necessity of extending the knowledge management perspective towards a network approach taking into account that value creation processes are increasingly conducted in networks of firms. The domain of self modelling knowledge network has therefore been introduced, describing the relevant tasks in a functional-decomposition diagram and the information objects in a semantic data model. The major advantage of the self modelling approach for KM is the fact that – if necessary – for every individual knowledge demand a knowledge network can be established. By doing so, a considerable contribution is made to the are of KM, which is still comparatively incomplete in the decentralised domain (c.f. [17, 18]). Based on the models and the concept of self modelling networks a component model for inter-organisational KM has been derived providing a basis for the implementation of an automated knowledge network system. The impetus was on developing a mechanism for the flexible acquisition and visualisation of knowledge and knowledge artefacts across several network nodes. The in-depth modelling of the knowledge demand was not focus of this paper. For this issue we refer to the results of specific research as described for example in [1, 5, 11, 15].

References

1. Abecker, A., et al., Toward a Technology for Organizational Memories. IEEE Intelligent Systems, 1998(May/June 1998): p. 40-48.
2. Albani, A., et al. Domain Based Identification and Modelling of Business Component Applications. in 7th East-European Conference on Advances in Databases and Informations Systems (ADBIS-03), LNCS 2798. 2003. Dresden, Deutschland: Springer Verlag.
3. Albani, A., et al. Component Framework for Strategic Supply Network Development. in 8th East-European Conference on Advances in Databases and Information Systems (ADBIS-04), LNCS. 2004. Budapest, Hungary: Springer Verlag.
4. Alpar, P. and D. Kalmring, Inter-Organizational Knowledge Management with Internet Applications. The 9th European Conference on Information Systems, 2001.
5. Apostolou, D., et al. Challenges and Directions in Knowledge Asset Trading. in PAKM 2002. 2002: Springer-Verlag.
6. Bazijanec, B., et al. A Component-based Architecture for Protocol Vector Conversion in Inter-organizational Systems. in International Workshop on Modeling Inter-Organizational Systems (MIOS'04). 2004. Larnaca: LNCS.
7. Canadas, G., et al., Framework for Automatic Generation of Ontology Mappings. Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento, 2004.
8. Carlsson, S.A., Knowledge Management and Knowledge Management Systems in Inter-organizational Networks. Knowledge and Process Management, 2003. **10**(3): p. 194-206.
9. Kalfoglou, Y. and M. Schorlemmer. Ontology Mapping: The State of the Art. in Semantic Interoperability and Integration. 2005.
10. Kreis-Hoyer, P. and J. Grünberg, Inter-Organizational Knowledge Networks: A Theoretical Foundation. 2002, European Business School.
11. Meisel, H. and E. Compatangelo. ConceptTool: Intelligent Support to Knowledge Management. in 7th International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES'2003). 2003. Oxford.
12. OMG, OMG Unified Modelling Language Spezification Version 2.0. 2003.
13. Riempp, G. Eine Architektur für integriertes Wissensmanagement. in WI 2003. 2003. Dresden.
14. Schmaltz, R. and S. Hagenhof, Wissensmanagement in unternehmensübergreifenden Kooperationen. 2003, Georg-August-Universität: Göttingen.
15. Schreiber, G., H. Akkermans, and A. Anjewierden, Knowledge Management and Engineering - The CommonKADS Methodology. 1999.
16. Seufert, A., A. Back, and G. van Krogh, Wissensnetzwerke: Vision-Referenzmodell-Archetypen und Fallbeispiele, in Wissensmanagement: Zwischen Wissen und Nichtwissen. 2000: München.
17. Tsui, E., Technologies for Personal and Peer-to-Peer (P2P) Knowledge Management, in CSC Leading Edge Forum (LEF) Technology Grant Report. 2002.
18. Tsui, E., Tracking the Role and Evolution of Commercial Knowledge Management Software, in Handbook on Knowledge Management, C.W. Holsapple, Editor. 2004: Berlin, Heidelberg. p. 5-27.
19. Turowski, K., ed. Standardized Specification of Business Components: Memorandum of the working group 5.10.3 Component Oriented Business Application System. 2002, University of Augsburg: Augsburg.

ORM 2005 PC Co-chairs' Message

Fact-Oriented Modeling is a conceptual approach to modeling and querying the information semantics of business domains in terms of the underlying facts of interest, where all facts and rules may be verbalized in language that is readily understandable by non-technical users of those business domains. Unlike Entity-Relationship (ER) modeling and UML class diagrams, fact-oriented modeling treats all facts as relationships (unary, binary, ternary etc.). How facts are grouped into structures (e.g. attribute-based entity types, classes, relation schemes, XML schemas) is considered a lower level, implementation issue that is irrelevant to the capturing essential business semantics. Avoiding attributes in the base model enhances semantic stability and populatability, as well as facilitating natural verbalization. For information modeling, fact-oriented graphical notations are typically far more expressive than those provided by other notations. Fact-oriented textual languages are based on formal subsets of native languages, so are easier to understand by business people than technical languages like OCL. Fact-oriented modeling includes procedures for mapping to attribute-based structures, so may also be used to front-end other approaches.

Though less well known than ER and object-oriented approaches, fact-oriented modeling has been used successfully in industry for over 30 years, and is taught in universities around the world. The fact-oriented modeling approach comprises a family of closely related "dialects", the most well known being Object-Role Modeling (ORM), Natural Language Information Analysis Method (NIAM), and Fully-Communication Oriented Information Modeling (FCO-IM). Though adopting a different graphical notation, the Object-oriented Systems Model (OSM) is a close relative, with its attribute-free philosophy.

Commercial tools supporting the fact-oriented approach include the ORM solution within Microsoft's Visio for Enterprise Architects, and the FCO-IM tool CaseTalk. Free ORM tools include InfoModeler and Infagon, as well as various academic prototypes. Dogma Modeler, an ORM-based tool for specifying ontologies, is currently being significantly extended. An open-source ORM2 tool is also currently under development to vastly extend both functionality and usability for the next generation of ORM. General information about ORM, and links to other relevant sites, may be found at http://www.orm.net/.

This year we had 26 submissions from all over the globe. After an extensive review process by a distinguished international program committee, with each paper receiving three or more reviews, we accepted the 14 papers that appear in these proceedings. Congratulations to the successful authors!

August 2005

Terry Halpin, Neumont University
Robert Meersman, Vrije Universiteit Brussel
(ORM'05 Program Committee Co-Chairs)

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, p. 602, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Using Abstractions to Facilitate Management of Large ORM Models and Ontologies

C. Maria Keet

Faculty of Computer Science, Free University of Bozen-Bolzano,
Piazza Domenicani 3, 39100 Bozen-Bolzano, Italy
phone +39 04710 161 28
keet@inf.unibz.it

Abstract. Due to ever larger ORM models and ORM-represented ontologies, information management and its GUI representation is even more important. One useful mechanism is abstraction, which has received some attention in conceptual modelling and implementation, as well as its foundational characteristics. Extant heuristics for ORM abstractions are examined and enriched with several foundational aspects of abstraction. These improvements are applicable to a wider range of types of representations, including conceptual models and ontologies, thereby not only alleviating the Database Comprehension Problem, but also facilitate conceptual model and ontology browsing.

1 Introduction

As a result of ever-increasing company size and complexity, the possibility increases to encounter the Database Comprehension Problem: the difficulty to understand and manage large conceptual models. Similarly, the size of ontologies increases (e.g. the Gene Ontology [28] contains about 18000 entities and the Foundational Model of Anatomy (FMA) [27] 72,000) and with ontology integration and formalisations, an Ontology Comprehension Problem emerges. Yet often only a *section* is of interest: a simplified higher level of granularity such as the GO slims [29], fact&entity finding where the query answer contains only the adjacent and high-level elements, or using a small selection of entities of an ontology when developing an ontology-inspired conceptual model, where zooming and abstraction simplifies the user's actions. Manual contextualisation of ontologies, e.g. with DOGMA [12] or C-OWL [2], can alleviate the problem of manageability and understandability, but they do not provide simplified views of the underlying complex model nor a 'zooming in'. One can apply levels of granularity to organise the conceptualisation, but it has to be pre-defined and is a static structure. In contrast, *abstraction* is the process to go from complex to simpler representations, which we focus on. Section 2 contains theoretical aspects of abstraction, in §3 assessed and compared with ORM abstraction heuristics as developed by Campbell et al [3] using different models and abstraction mechanisms. We discuss results, integrate it with the theoretical aspects of abstraction, and propose improvements – reordering rules, maintaining rules 1-6,12, and replacing 7-11

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 603–612, 2005.
© Springer-Verlag Berlin Heidelberg 2005

with the generic abstractions introduced in this article – that are more widely applicable to conceptual models and ontologies for different purposes (§4). We finalise with some concluding remarks.

2 Abstractions

Two meanings of abstraction are common in the literature, which are the duality between abstraction and concretisation and abstraction by the process of ignoring details or the bigger picture. We focus on the second meaning: how to *not* take into account things that are not of interest. This is different from indistinguishability in that with the latter we *cannot* observe a difference at a certain level, whereas with abstraction we *choose to disregard* undesired aspects. What the undesired aspects are and how to ignore them depends on i) the subject domain, ii) user's perspective and context, iii) the type of abstraction, iv) the procedure of (consecutive steps of) abstraction, and v) on what type of representation/model the abstraction is performed. We focus on point iii-v, although i-ii does influence possible usage and solutions proposed in the extant literature.

Manual efforts of abstraction has been, and is being, carried out informally or in somewhat structured fashion with UML modelling, (E)ER (e.g. [11]) and the Abstraction Hierarchy (AH, e.g. [18] [24]), which are laborious and intuitive ad hoc methods. AH does not even have a supporting modelling paradigm such as UML and (E)ER and is akin the 'complete freedom' biologists have with their "black boxes" (e.g. Physiome Project [10], among many). Also ecological modelling tools, such as STELLA/*ithink* [31], still maintain relative freedom to abstract, with the consequential manual effort it requires to do so (e.g. [23] [14]). A different abstraction approach that does not suppress the details, but abstracts away that what is deemed less important because it is *non-functional* [8], is not elaborated on further here. Also, the reductionism versus holism & systems biology where the former ignores the larger systems and simplifies to its smaller (sub-)components, is left for another occasion.

The remainder of this section discusses approaches to abstraction that can be carried out automatically. Abstractions based on heuristics have been developed by [3], who aimed to simplify large ORM models by suppressing roles and objects that, based on the encoded semantics, are less relevant. This is discussed in detail in sections 3 and 4. Ghidini and Giunchiglia [6] formalised abstraction by exploiting Local Model Semantics of context reasoning and formalising abstraction "as a pair of (formal) languages plus a mapping between them", using the *abs* function for syntax mapping. Given the ground language L_0 and the abstract language L_1, symbol abstraction operates on constant symbols (comparable to ORM objects): $c_1, ..., c_n \in L_0, c \in L_1$ and $abs(c_i) = c$, for all $i \in [1, n]$. Idem ditto for function symbols with $abs(f_i) = f$ and predicate symbols $abs(p_i) = p$, where arity abstractions on functions and predicates lower the arity by one; if the arity in L_0 is 1, f_i maps into a constant and p_i into a proposition [6]. What this *abs* function actually does, is, given an entity A, to return the parent entity B that subsumes A. Using specialisation/generalisation as abstraction is a recurring theme across domains and modelling paradigms [5] [4] [21] [13] [9].

Similar to the abstraction mechanism that takes advantage of the isA relationship, is abstraction through the partOf relationship, which is not based on set theory but mereology. While this is an ontological distinction, this can be implemented set-wise by giving in a little to the ontological trade-off. In essence, through abstraction the parts that make the whole are abstracted exploiting the partOf relation between the entities involved [1] [16] [17]. It is a point of philosophical debate if sub-processes are (a special kind of) partOf of its grander process [20] or involvedIn the grander process [16]. For example that 'I go from Rome to Bolzano' as parent process and take first the bus, then the metro, finally catch the train, and reading a book and listening to music at the same time. For purpose of clarity, we indicate these kind of more detailed sub-processes that can be abstracted away with $involvedIn(x, y)$. In an ontology, x and y are both perdurants, but in a conceptual model they are likely to occur as fact types, relations or methods.

Mani [19] introduced four types of abstraction, in addition to "type shifting operators" for grain size shifts. Type shifting goes from coarse to finer-grained with event to processes, process to states, and process to objects, and three more operations to fold processes and states and to fold events and propositions, preserving compositionality of two logical forms that are abstracted. This, then, is combined with the type shifting operators to create three non-endocentric abstractions and three endocentric (meronymic) abstractions. Pandurang and Levy [21] also emphasise compositionality and use a two-step process 1) abstraction of the intended domain model and 2) define set of formulas that formalises the abstracted domain to make the simplifying assumptions explicit in the base (more detailed) model. Like Ghidini and Giunchiglia [6], Pandurang and Levy [21] exploit the *isA* relation for abstraction. Although Mani's family of abstraction functions is developed for dealing with polysemy and underspecification in linguistics, it is a more promising approach than [6] because it captures the varying semantics of abstraction better than [6]'s pure syntactic approach. However, developing a computational implementation of the folding operations may not be easy. Multiple types of abstraction functions can be useful in particular for abstracting biological complex entities like *Second messenger system* or *MAPK signalling*. The former collapses processes such as *Activation*, *GTP-GDP exchange*, *α-subunit release*, states like *activated*, and components such as *Hormone receptor*, G_s *protein*, and *cAMP*. *MAPK signalling* is already used as a module in systems biology that at a higher level of abstraction is treated as a black box, containing (sub-)processes, input/output behaviour, parameters and their values, etc. [22]. Summarising the different usages, and options, for performing an abstraction, one can identify abstraction based on:

1. Taxonomy:moving 'up'in the specialisation/generalisation hierarchy through the isA relation, abstracting away a distinguishing property.
2. Partonomy (mereology): through the partOf relation.
3. ORM heuristics.
4. UML modularisation.
5. (E)ER abstractions.

6. The black boxes in biology, ecology models, and Abstraction Hierarchy.
7. Folding operations of different types of entities resulting in other types (perdurant into endurant etc), focussed on linguistics.
8. Syntax limited to Local Model Semantics of contextual reasoning.

Grouping these methods of abstraction into *types* of operation, then (1,2) focuses on exploiting primitive relations (relations like causality and aggregation are omitted because it is beyond the current scope or covered with existing rules (see [7]), (1-3, 5, 7) takes into consideration only the 'up' direction to a simplified level. Methods (3-5) are motivated by the database community to keep conceptual models comprehensible and manageable and (6, 7) are UoD-motivated – (1-5, 8) are domain-independent – although (3-7) all are bottom-up driven. Number (8) is syntax-based, where (3, 5) still take into account the coded semantics and thereby seem 'closer' to the domain-centred (informal) abstractions of (6, 7) than the syntax-approach of (8). Last, (1-3, 8) can, in principle, be carried out automatically without user intervention.

3 Experimentation

3.1 Methodology

The aim of the experiments is to assess Campbell et al's [3] abstraction heuristics with two distinct ORM models, and compare this with other abstraction methods.

Bacteriocins is most similar to Campbell et al's case study model because it also is an ORM model for a database (developed and in use; diagram omitted due to space limitations). On the other hand, the *Blood* ORM model (Fig.2) is a resultant of positioning orthogonally the relevant sections of the partonomy and taxonomy of the FMA [27], the Mode of Transmission perspective of infectious diseases [15], and a section of microorganism phylogeny. A similar model to *Blood* can be constructed also with *Bacteriocins* by linking it with the Gene Ontology [28], Agricultural Ontology Services [25], MetaCyc [32], SNOMED-CT [34], Bad Bug Book [26], and microorganism phylogeny. The test procedure is as follows: **A)** Take *Blood*, respectively *Bacteriocins*, and use Campbell's abstraction heuristics. **B)** Compare the result with the result based on a manual semantics analysis. Test where any of the logical and/or ontologically founded, theoretical abstraction mechanisms (points 1, 2, 7, 8 in the previous section) can be useful for improving procedure and outcome. **C)** Test other abstraction mechanisms: **C1** *UML-like modularisation*: group orthogonal sections (Anatomy, InfectiousAgent, ModeOfAction, blood process). **C2** *Modified abstraction heuristics*, taking into account above-mentioned domain-independent abstraction mechanisms: 1) Group anatomy isA: intermediate ones and leaves that are not involved in any role; 2) Group anatomy partOf: remove intermediates and leaves that are not involved in anything else than partonomy; 3) Remove non-mandatory leaves, roles, and 'dead ends' (successive leaves and self-contained subsections). 4) Reapply steps 1-3, if applicable. **D)** Apply C2 to Campbell's case study model.

3.2 Results

Before applying any heuristics, we abstracted *Blood* intuitively, based on semantics alone (Fig.1-D). Table 1 contains the summarised ORM heuristics rules taken from [3], with their applicability to the *Blood* and *Bacteriocins* ORM models. A salient aspect of the rules is the emphasis on treatment of role-set

Table 1. Abstraction rules and applicability to ORM models *Blood* and *Bacteriocins*

ORM heuristics	***Blood***	***Bacteriocins***
Rule 1: mandatory roles, 10 points	+	+
Rule 2: unary roles, 10 points	N/A	N/A
Rule 3: non-leaf object types, 9 points (thus, delete leaves)	+	+
Rule 4: smallest maximum frequency, 8 for uniqueness constraint, else lower	+	+
Rule 5: non-value types, 7 points (thus, value types have lower importance)	N/A	+
Rule 6: anchor points, 6 points	+	+
Rule 7: single-role set constraint, 5 points	N/A	N/A
Rule 8: multi-role set constraints, 4 points	N/A	N/A
Rule 9: set constraints and anchor points, 3 points	N/A	N/A
Rule 10: joining roles of set constraints, 2 points	N/A	N/A
Rule 11: first role of set constraint, 1 point	N/A	N/A
Rule 12: first role of internal uniqueness constraint, 1 point	+	+

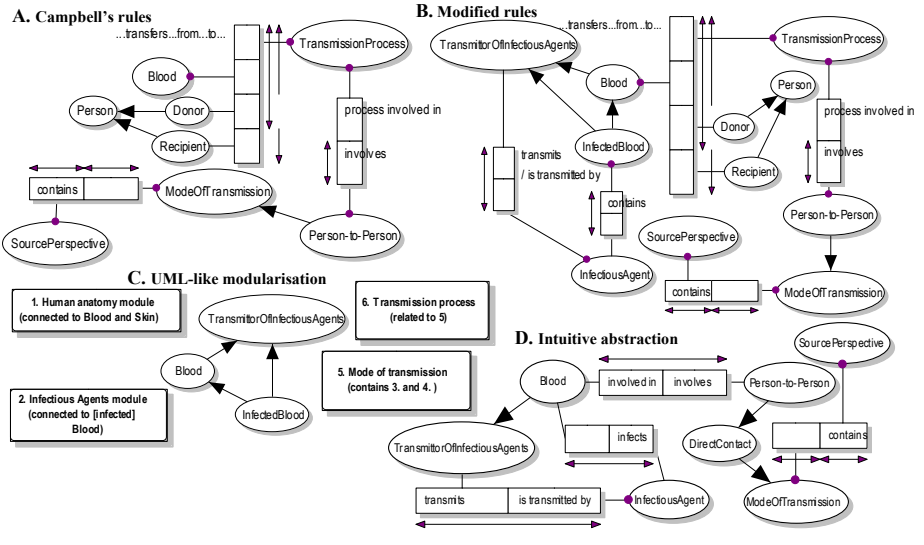


Fig. 1. A: *Blood* with Campbell's rules after 7 steps; B: *Blood* with adapted rules after 3 steps; C Manual UML-like modularisation after 3 steps (modules 3,4 abstracted within 5); D: semantic abstraction

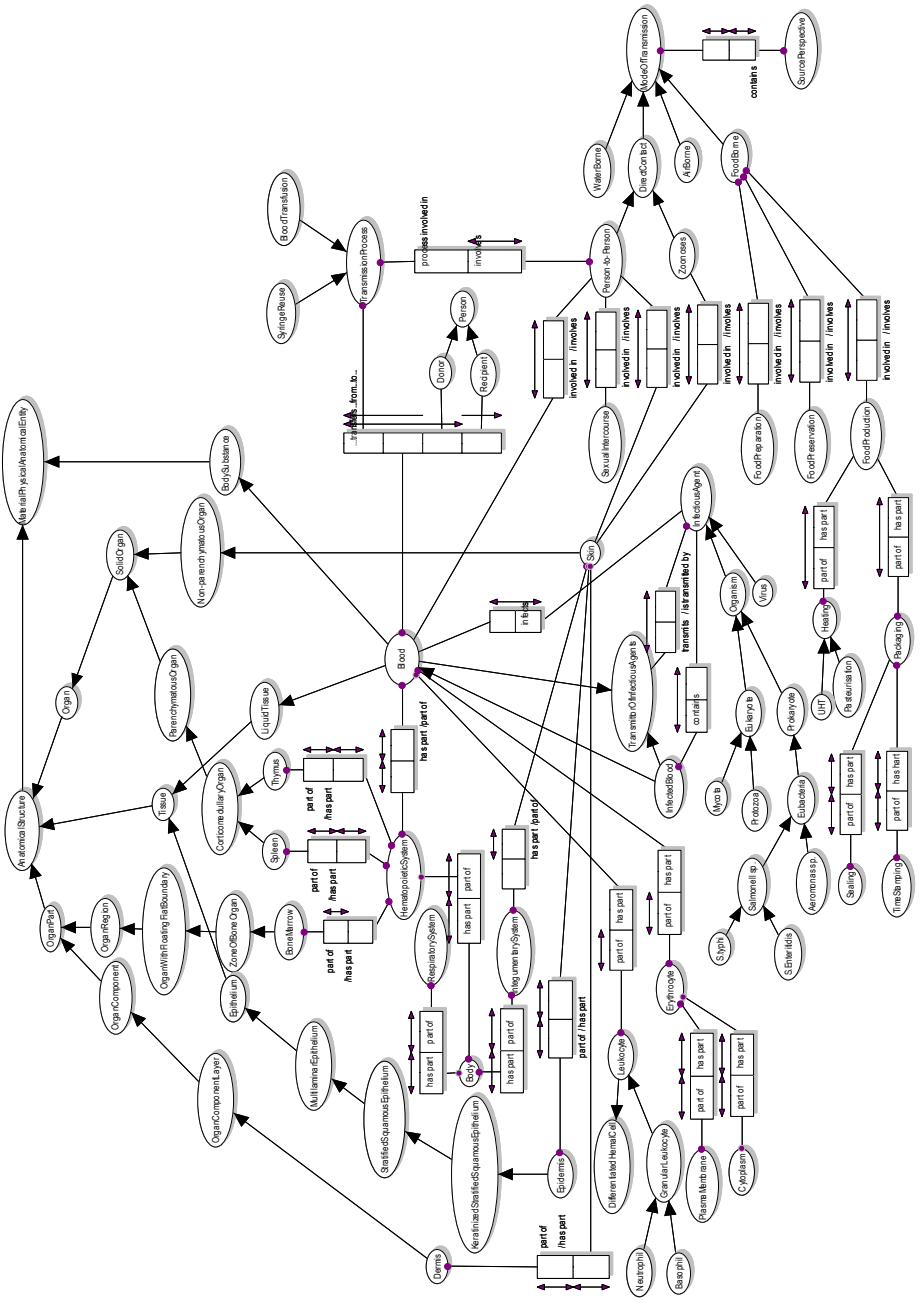


Fig. 2. The *Blood* ORM model showing the intersection of different pespectives (anatomy, phylogeny, mode of transmission) and processes of blood

constraints, which, if absent as in *Blood* and *Bacteriocins*, limits the 12 rules to only 7. The weight allocation to the rules [3] were adhered to, to the extend of counting which object type had most mandatory roles and uniqueness constraints, and for other rules used in descending order. Fig.1-A shows the result after 7 iterations of *Blood* with Campbell's heuristics. With the modified rules as listed in C2 in §3.1, then a near identical abstraction emerges after 3 iterative abstractions, with the added advantage that the infection semantics is still present (Fig.1-B); after the 4^{th} iteration it is identical to Fig.1-A. The UML-like abstraction result (Fig.1-C) has the neat aspect that it compartmentalises into the granular perspectives it was originally built up from, thereby correctly closing the circle. Testing *Bacteriocins* with Campbell's rules (see Table 1), it achieved maximum abstraction after 3 iterations. The set constraints rules were left unused, but for *Bacteriocins*, rules for collapsing subsumption and partOf relations would not have been of use either, unless if *Bacteriocins* had been extended analogous to *Blood*. Applying C2 to Campbell et al's case study model, the same final abstracted model was achieved after two iterations of the rules.

4 Discussion

Campbell et al's rules focus on semantic importance *assumed* from syntax and implying that what is not meant in the rules must be unimportant. However, for instance, a mandatory role does not *imply* semantic importance in the UoD, nor does it appear in foundational aspects of abstraction: the *Disease*-related mandatory fact and types in *Bacteriocins* are important to its immediate neighbours, but in the overall semantics play only a secondary role; according to the client, it had the status of peripheral nice aspect. Likewise, *FoodBorne* is irrelevant for person-to-person transmission in *Blood*. No model captures this and relying on encoded semantics will not address it either. On the contrary, if one knows the model, abstraction heuristics are influenced by such background information. Campbell et al's case study ORM model contains several role set-related rules, which are not necessarily a salient feature in ORM models and certainly not in other representation methods. If *Blood* and *Bacteriocins* would have been used as first case study instead of Campbell et al' case study, neither rules (7-11) nor the unary and mandatory ones would have been included as such, although most aspects of the remaining rules are, heuristically, useful. This illustrates limitations of bottom-up case study based approach: generalising from a bottom-up approach can require tweaking a 'generalised reusable' theory to make it more general. In an effort not to 'pollute' rules so that they only would fit the ORM case study models we brought in, theoretical notions of abstraction are useful and beneficial to improve heuristics and make them applicable to a wider range of models. Taking into account §2, rules, and automation, let x, y, and z be entities (object types, perdurants/endurants), y is at a higher level of abstraction, z related to x, and *abs* a function, then

1. isA and partOf collapsing:

$$\text{if } isA(x, y), \text{ then } abs_{isA}(x) = y \tag{1}$$

$$\text{if } partOf(x,y), \text{ then } abs_{partOf}(x) = y \tag{2}$$

$$\text{if } isA(x,y) \wedge \neg relatedTo(x,z), \text{ then } abs_{isA}(x) = y \tag{3}$$

$$\text{if } partOf(x,y) \wedge \neg relatedTo(x,z), \text{ then } abs_{partOf}(x) = y \tag{4}$$

The reason to include $\neg relatedTo(x,z)$ in (3-4) is to ensure abstracting the hierarchy by removing only intermediate layers that serve no other purpose; this is the same as Campbell's et al's "lowest common identified supertype". Note that for abstracting taxonomies and partonomies, (3-4) is of no use but it certainly is for conceptual models and for (more complex) formal ontologies. In ORM, partOf and its inverse hasPart are merely roles in a fact type, therefore an implementation either will need an additional (text string) analysis of role names, or become a separate element.

2. Additional semi-automation of predicates and role-set constraints inspired by [6]'s syntax approach. Then, in addition to $abs_{isA}(x)$ for taxonomic subsumption, for any predicate p, we have

$$\text{if } p(x,y), \text{ then } abs_{p}(x) = y \tag{5}$$

3. For processes and sub-processes a clear distinction has to be made between *subsumption* of processes for which $abs_{isA}(x)$ suffices, and sub-processes of the $involvedIn(x,y)$ type with the abstraction rule

$$\text{if } involvedIn(x,y), \text{ then } abs_{in}(x) = y \tag{6}$$

4. Complex folding, where w is a *new* object type added to the model

$$(abs_{isA}(x_1) \wedge ... \wedge abs_{isA}(x_n) \wedge abs_{p}(z_1) \wedge ... \wedge abs_{p}(z_n)) = w \tag{7}$$

Note that if one were to allow such operation, compositionality is not maintained, because it requires introduction of a new entity w. It is possible provided a higher abstraction level has been defined in an ontology and computability of the abstraction operation is required (e.g. [21]). It can function as simpler engineering solution than Mani's folding family in order to abstract, for instance, the *Second messenger system*. However, user intervention may be preferred.

5. UML-like modularisation for to modularise-able perspectives, AHs, and black boxes in biology and ecology.

Are (7), point 5, and its envisaged implementations less straightforward because it may be too challenging to capture in simple rules the complex 'flexibility' of human-performed abstraction? Businesses and their models are developed by humans – nature is not. For a biology UoD, one can resort to manual modularisation, combine it with a static structure of pre-defined granularity and use abstraction to move from contents in one level to the other, or firing a combination of rules like (7) to approximate the complexity. With the latter, one can set up separate decision trees, whereby a tool like iCOM [30] may be useful because

it already contains a (Description Logics) reasoner. This will also facilitate scaling up experimentation and varying weights given to rules to make these less dependent on only a few (ORM) models. Regardless automation of abstraction, changing the extant heuristics to better reflect semantic and syntactic richness of (ORM) models facilitates wider applicability. The proposed change in rules, in particular for ORM conceptual models, is to maintain rules 1-6,12 and replacing 7-11 with the generic abstractions introduced in this paragraph. Using the generic abstractions *only* or *before* the ORM rules can reduce the amount of iterations, in particular if (some of) the model uses (sections of) ontologies. In addition, the generic rules are useful for *both* conceptual models *and* ontologies, thereby extending the potential for consistent reuse of abstraction across different types of knowledge representations.

5 Conclusions

Existing ORM abstraction heuristics to simplify large models were examined with two distinct types of ORM models and heuristics augmented with foundational aspects of abstraction. This improvement is applicable to a wider range of types of knowledge representations, thereby has the potential to be of use not only to alleviate the Database Comprehension Problem, but also conceptual model and ontology browsing. Further experimentation with other models, automation, and possible combinations of rules with algorithms are needed if it is to perform effective abstraction of the more complex abstraction mechanisms of 'folding multiple entities' as carried out in the biology domain.

References

1. Bittner, T., Smith, B.: A Theory of Granular Partitions. In: Foundations of Geographic Information Science, Duckham, M, Goodchild, MF, Worboys, MF (eds.), London: Taylor & Francis Books (2003) 117-151
2. Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., Stuckenschmidt, H.: Contextualizing Ontologies. Journal of Web Semantics (2004) 1(4):24
3. Campbell, L.J., Halpin, T.A. and Proper, H.A.: Conceptual Schemas with Abstractions: Making flat conceptual schemas more comprehensible. Data & Knowledge Engineering (1996) 20(1): 39-85
4. Degtyarenko, K., Contrino, S. COMe: the ontology of bioinorganic proteins. BMC Structural Biology, (2004) 4:3.
5. Fonseca, F., Egenhofer, M., Davis, C., and Camara, G.: Semantic Granularity in Ontology-Driven Geographic Information Systems. Annals of Mathematics and Artificial Intelligence (2002) 36 (1-2): 121-151
6. Ghidini, C., Giunchiglia, F.: A semantics for abstraction. Technical Report DIT-03-082, University of Trento, Italy (2003)
7. Halpin, T.: Information Modeling and Relational Databases. San Francisco: Morgan Kaufmann Publishers (2001)
8. Hanahan, D., Weinberg, R.A.: The Hallmarks of Cancer. Cell (2000) 100: 57-70
9. Hobbs, J.R.: Granularity. International Joint Conference on Artificial Intelligence (IJCAI85) (1985) 432-435

10. Hunter, P.J., Borg, T.: Integration from Proteins to Organs: The Physiome Project. Nature (2003) 4(3): 237-243
11. Jaeschke, P., Oberweis, A., Stucky, W.: Extending ER Model Clustering by relationship clustering. 12th International Conference on Entity Relationship Approach, Arlington, Texas (1993)
12. Jarrar, M., Demy, J., Meersman, R.: On Using Conceptual Data Modeling for Ontology Engineering. Journal on Data Semantics Special issue on 'Best papers from the ER/ODBASE/COOPIS 2002 Conferences' (2003) 1(1): 185-207
13. Kiriyama, T., Tomiyama, T.: Reasoning about Models across Multiple Ontologies. International Qualitative Reasoning Workshop (1993)
14. Keet, C.M.: Factors affecting ontology development in ecology. In: Data Integration in the Life Sciences (DILS2005), Ludäscher, B., Raschid, L. (Eds.), Springer LNBI 3615, (2005) 46-62
15. Keet, C.M., Kumar, A.: Applying partitions to infectious diseases. XIX International Congress of the European Federation for Medical Informatics. Geneva, Switzerland (2005)
16. Kumar, A., Smith, B., Novotny, D.D.: Biomedical Informatics and Granularity. Comparative and Functional Genomics (2005) 5(6-7): 501-508
17. Kumar, A., Yip, L., Smith, B., Grenon P.: Bridging the Gap between Medical and Bioinformatics Using Formal Ontological Principles. Computers in Biology and Medicine (In press)
18. Lind, M.: Making sense of the abstraction heirarchy. Cognitive Science Approaches to Process Control (CSAPC99), Villeneuve d'Ascq, France 21-24 September (1999)
19. Mani, I.: A theory of granularity and its application to problems of polysemy and underspecification of meaning. In: Principles of Knowledge Representation and Reasoning (KR98), A.G. Cohn, L.K. Schubert, and S.C. Shapiro (eds.). San Mateo: Morgan Kaufmann (1998) 245-255
20. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: Ontology Library. WonderWeb Deliverable D18 (v1.0). http://wonderweb.semanticweb.org (2003)
21. Pandurang Nayak, P., Levy, A.Y.: A semantic theory of abstractions. In: Proceedings of the International Joint Conference on Artificial Intelligence, Mellish, C. (ed.). San Mateo: Morgan Kaufmann (1995) 196-203
22. Sontag, E.D.: Some new directions in control theory inspired by systems biology. Systems biology (2004) 1(1): 9-18
23. Tett, P., Wilson, H.: From biogeochemical to ecological models of marine microplankton. Journal of Marine Systems, 25 (2000) 431-446
24. Yu, X, Lau, E., Vicente, K.J., Carter, M.W. Toward theory-driven, quantitative performance measurement in ergonomics science: the abstraction hierarchy as a framework for data analysis. Theoretical Issues in Ergonomics Science (2002) 3(2): 124-142
25. Agricultural Ontology Services. http://www.fao.org/agris/aos
26. Bad Bug Book. http://www.cfsan.fda.gov/ mow/intro.html
27. Foundational Model of Anatomy. 2003. http://sig.biostr.washington.edu/projects/fm/index.html
28. Gene Ontology Consortium. http://www.geneontology.org
29. GO-slim. http://www.geneontology.org/GO.slims.shtml
30. iCOM. http://www.inf.unibz.it/∼franconi/icom
31. ISEE Systems. http://www.iseesystems.com
32. MetaCyc & BioCyc. http://BioCyc.org
33. Open Biological Ontologies. http://obo.sourceforge.net
34. Snomed CT. http://www.snomed.org/snomedct/

Modularization and Automatic Composition of Object-Role Modeling (ORM) Schemes

Mustafa Jarrar

Vrije Universiteit Brussel, Brussels, Belgium
mjarrar@vub.ac.be

Abstract. In this pape0072 we present a framework and algorithm for modularization and composition of ORM schemes. The main goals of modularity are to enable and increase reusability, maintainability, distributed development of ORM schemes. Further, we enable effective browsing and management of such schemes through libraries of ORM schema modules. For automatic composition of modules, we present and implement a composition operator: all atomic concepts and their relationships (i.e. fact-types) and all constraints, across the composed modules, are combined together to form one schema (called modular schema).

Keywords: Object Role Modeling, ORM, NIAM, Conceptual Modeling, Ontology, Formal ontology engineering, DOGMA, DogmaModeler, Modularization, Composition, Reusability, Distributed Development, Maintainability.

1 Introduction and Motivation

ORM (Object-Role Modeling) [H01] is a conceptual modeling approach that was developed in the early 70's. It is a successor of the NIAM (Natural-language Information Analysis Method) [VB82]. The ORM conceptual schema methodology is fairly comprehensive in its treatment of many "practical" or "standard" business rules and constraint types (e.g. identity, mandatory, uniqueness, subsumption, subset, equality, exclusion, value, frequency, symmetric, intransitive, acyclic, etc.). Furthermore, ORM has an expressive and stable graphical notation since it captures many rules graphically and it minimizes the impact of change on the models.

Although ORM was originally developed as a database modeling approach, it has been also successfully reused in other conceptual modeling scenarios, such as XML-Schema conceptual design [BGH99], business rule modeling language [H04][N99][DJM02a], ontology modeling [JDM03][J05], etc. Hence, we shall regard an ORM schema, in this paper, as a general conceptual model independently of a certain application or modeling scenario; and we sometimes interchange the term "ORM schema" with the term "axiomatization" to refer to the same thing.

The main idea of ORM modularization in this paper is to decompose an ORM schema into a set of smaller related modules, which: 1) are easier to reuse in other kinds of applications; 2) are easier to build, maintain, and replace; 3) enable distributed development of modules over different locations and expertise; 4) enable

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 613 – 625, 2005.
© Springer-Verlag Berlin Heidelberg 2005

the effective management and browsing of modules, e.g. enabling the construction of libraries of ORM modules [JM02b][1].

To compose modules automatically, we propose a composition operator: all atomic concepts and their relationships (i.e. fact-types) and all constraints, across the composed modules, are combined together to form one schema (called *modular schema*).

1.1 A Simple Example

In what follows, we give an example to illustrate the (de)composition of ORM schemes. Fig. 1 shows two ORM schemas of Book-Shopping and Car-Rental applications. Notice that both schemes share the same axioms about payment.

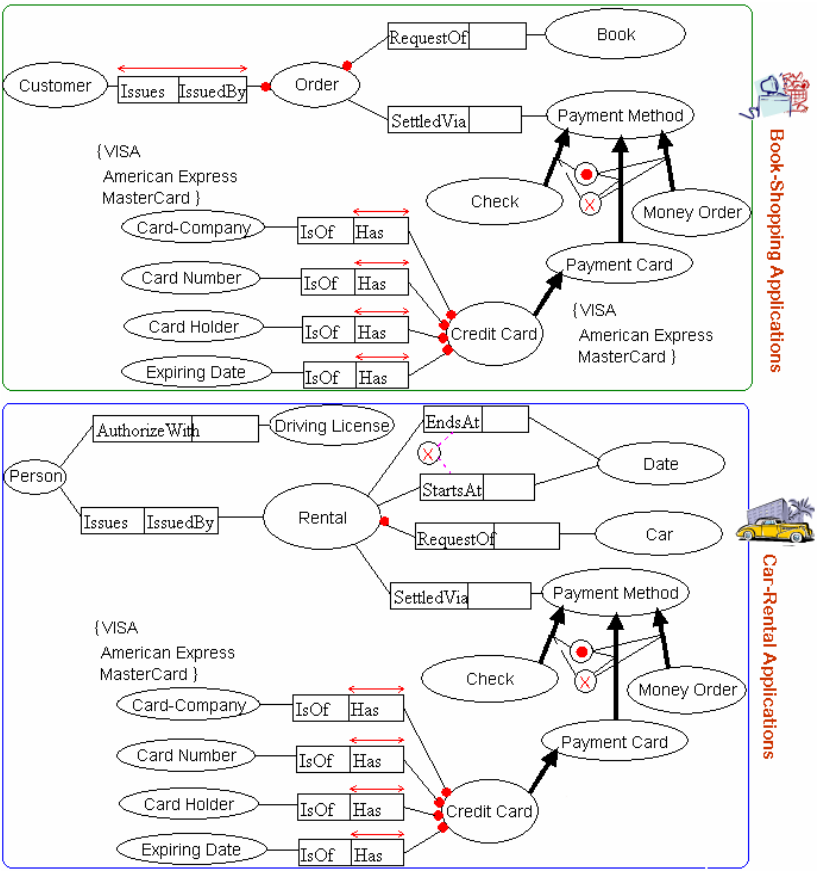


Fig. 1. Book-shopping and Car-Rental schemes

[1] Notice that reusability, maintainability, and distributed development of ORM schemes might not be challenges in database modeling (the original usage of ORM), but they are urgent demands when using ORM e.g. in ontology Engineering [J05][SMS+05].

Instead of repeating the same effort to construct the axioms of the "payment" part, we suggest decomposing these schemes into three modules, which can be shared and reused among other applications (see fig. 2). Each application-type (*viz.* Book-Shopping and Car-Rental) selects appropriate modules (from a library) and composes them through a composition operator. The result of the composition is seen as one schema[2].

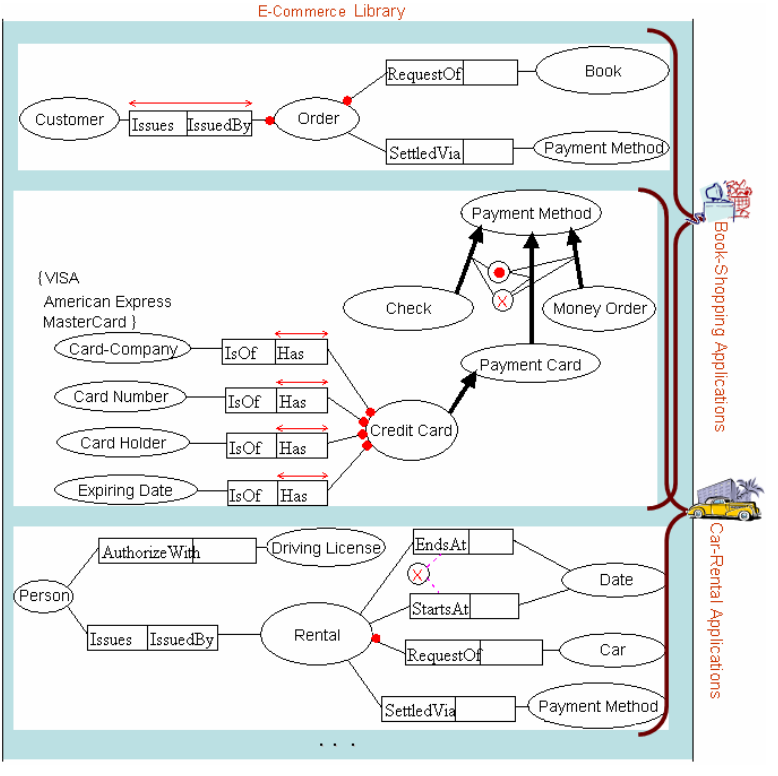


Fig. 2. Modularized schemes

Engineering schemes in this way will not only increase their reusability, but also the maintainability and management of these axiomatizations[3]. As the software engineering literature teaches us, small modules are easier to understand, change, and replace [P72] [SWCH01]. An experiment by [BBDD97] proves that the modularity of object-oriented design indeed enables better maintainability and extensibility than structured design. Decomposing schemes into modules also enables the distributed development of these modules over different location, expertise, and/or stakeholders.

[2] The illustrated composition in this example is very simplistic, as each pair of modules overlap only in one object-type, i.e. the "Payment Method". In farther sections, we discuss more complicated compositions, in which rules in different modules may contradict or imply each other.

[3] In this way, one can imagine axiomatizations (/schemes) as large sets of business rules modularized and organized as sets of compose-able modules.

As an analogy, compare the capability of distributing the development of a program built in Pascal with a program built in JAVA, i.e. structured verses modular distributed software development.

The modularity criteria could typically be subject-oriented and/or purpose/task-oriented. Subject-oriented parts should be released into separate modules, e.g. separate between the financial axioms (e.g. salary, contract, etc.) and the academic axioms (e.g. course, exams, etc.). The general purpose/task-oriented parts of an axiomatization could be released into separate modules, e.g. the axiomatization of "payment", "shipping", "invoicing", which are often repeated in many e-commerce applications.

2 Composition Framework

To compose modules we define a composition operator. All concepts and their relationships (i.e. fact-types) and all constraints, across the composed modules, are combined together to form one axiomatization. In other words, the resultant composition is the union of all axioms in the composed modules. As shall be discussed later, a resultant composition might be *incompatible* in case this composition is not satisfiable, e.g. some of the composed constraints might contradict each other.

Our approach to composition is constrained by the following argument. A developer, when including a module into another, must expect that all rules in the included module are inherited by the including module, i.e. *all axioms in the composed modules must be implied in the resultant composition*. Formally speaking, the set of possible models for a composition is the intersection of all sets of possible models for all composed modules. In other words, we shall be interested in the set of models that satisfy all of the composed modules.

In fig. 3, we illustrate the set of possible instances (i.e. possible models) for a concept constrained differently in two modules composed together. Fig. 3(a) shows a compatible composition where the set of possible instances for M.c is the intersection of the possible instances of M_1.c and M_2.c. Fig. 3(b) shows a case of incompatible composition where the intersection is empty.

Notice that our approach to module composition is not intended to integrate or unite the extensions (i.e. ABoxes) of a given set of modules, as several approaches to

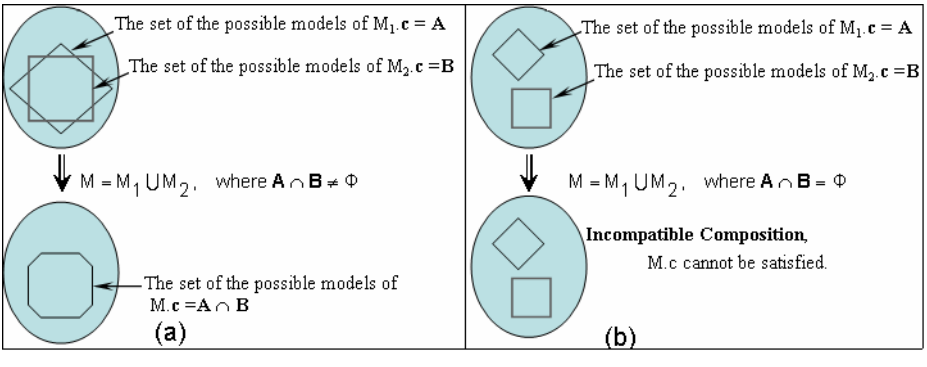


Fig. 3. (a) Compatible composition, (b) Incompatible composition

schema integration aim to do [SP94] [SK03][BS03]. *Our concern is to facilitate developers (at the development phases) with a tool to inherit (or reuse) axiomatizations without "weakening" them*. In other words, when including a module into another module (using our composition operator, which we shall formalize in the next sections) all axioms defined in the included module should be inherited by (or applied in) the including module.

It is also worth to mention that Vermeir [V83] has proposed an approach for modularizing large ORM diagrams based on heuristic procedures. However, this approach is not related to ours, as it is only concerned with how to "view" a one large ORM diagram in different degrees of abstraction or viewpoints. Another similar approach is proposed by Shoval [S85].

2.1 Definition (Module)

A module is an axiomatization (i.e. a typical ORM Schema) of the form $M = \langle \mathbf{P}, \mathbf{\Omega} \rangle$, where $\mathbf{P}$ is a non empty set of fact-types, i.e. the set of object-types and their relationships; $\mathbf{\Omega}$ is a set of constraints which declares what should necessarily hold in any possible world of M. In other words $\mathbf{\Omega}$ specifies the legal models of M.

2.2 Definition (Model, Module Satisfiability)

Using the standard notion of an interpretation of a first order theory, an interpretation I of a module M, is a *model* (also called "legal model") of M iff each sentence of M (i.e. each $\rho \in P$ and each $\omega \in \Omega$) is true for I.

Each module is assumed to be self-consistent, i.e. satisfiable. Module satisfiability demands that each role in the module can be satisfied [BHW91]. For each ρ in a given module M, ρ is satisfiable w.r.t. to M if there exists a model I of M such that $\rho^I \neq \varnothing$.

Notice that we adopt a strong requirement for satisfiability, as we require each role in the schema to be satisfiable. A weak satisfiability requires only the module itself (as a whole) to be satisfiable [H89][BHW91].

2.3 Definition (Composition Operator)

Modules are composed by a composition operator, denoted by the symbol '$\oplus$'. Let $M = M_1 \oplus M_2$, we say that M is the composition of M_1 and M_2. M typically is the union of all fact-types and constraints in both modules. Let $M_1 = \langle \mathbf{P}_1, \mathbf{\Omega}_1 \rangle$ and $M_2 = \langle \mathbf{P}_2, \mathbf{\Omega}_2 \rangle$, the composition of $(M_1 \oplus M_2)$ is formalized as $M = \langle \mathbf{P}_1 \oplus \mathbf{P}_2, \mathbf{\Omega}_1 \oplus \mathbf{\Omega}_2 \rangle$. A composition $(M_1 \oplus M_2)$ should *imply* both M_1 and M_2. In other words, for each model that satisfies $(M_1 \oplus M_2)$, it should also satisfy each of M_1 and M_2. Let $(M_1)^I$ and $(M_2)^I$ be the set of all possible models of M_1 and M_2 respectively. The set of possible models of $(M_1 \oplus M_2)^I = (M_1)^I \cap (M_2)^I$. A composition is called *incompatible* iff this composition cannot be satisfied, i.e. there is no model that can satisfy the composition, or each of the composed modules.

2.4 Definition (Modular Schema)

A modular schema $M = \{M_1 \ldots M_n, \oplus\}$ is a set of modules with a composition operator between them, such that $P = (P_2 \oplus \ldots \oplus P_n)$ and $\mathbf{\Omega} = (\mathbf{\Omega}_1 \oplus \ldots \oplus \mathbf{\Omega}_n)$.

3 Composition of ORM Conceptual Schemes

In this section we present an algorithm for automatic composition of modules specified in ORM. We adopt the ORM formalization and syntax as found in [H89][H01], excluding three things. First, although ORM supports n-ary predicates, only binary predicates are considered in our approach. Second, our approach does not support objectification, or the so-called nested fact types in ORM. Finally, our approach does not support the derivation constraints that are not part of the ORM graphical notation.

A composition of two modules ($M = M_1 \oplus M_2$) is performed in the following steps: 1) Combine the two sets of fact types ($\mathbf{P} = \mathbf{P}_1 \oplus \mathbf{P}_2$). 2) Combine the two sets of constraints, $\mathbf{\Omega} = \mathbf{\Omega}_1 \oplus \mathbf{\Omega}_2$. 3) Reason to find out whether the composition is satisfiable. Optionally, 4) reason to eliminate all implied constraints from the composition. The last two steps are not presented in this paper because of the limited space. See our approach in [JH05] for reasoning about the satisfiability of ORM Schemes. For step 4 we refer to [H89] for a comprehensive specification of constraint implication in ORM.

The composition is considered an incompatible operation (and thus terminated) iff the result cannot be satisfied.

Step 1: Combining fact types

When composing two sets of fact-types ($\mathbf{P} = \mathbf{P}_1 \oplus \mathbf{P}_2$), an object-type $M_1(T)$ in module M_1 and a object-type $M_2(T)$ in module M_2 are considered exactly the same concept iff they are referred to by the same term T, and/or URI. Formally, ($M_1(T) = M_2.(T)$). Likewise, two fact-types are considered exactly the same ($M_1.<T_1, r, r', T_2> =$

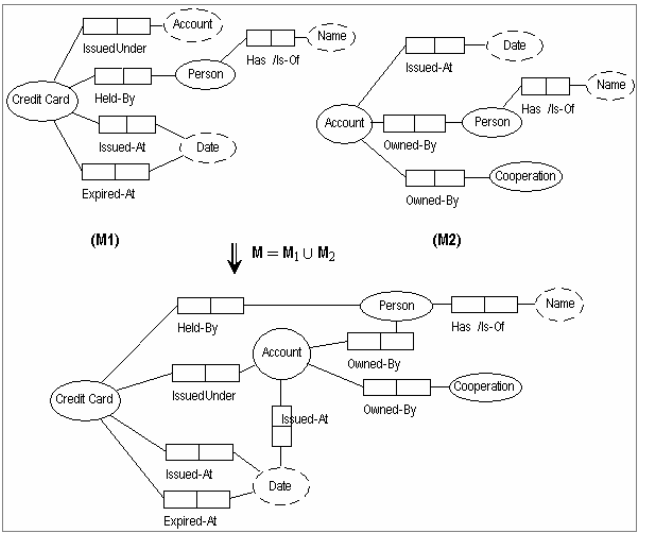


Fig. 4. Combining ORM fact types

$M_2.<T_1, r, r', T_2>$) iff $M_1(T_1) = M_2(T_1)$, $M_1.(r) = M_2.(r)$, $M_1.(r') = M_2.(r')$, and $M_1.(T_2) = M_2.(T_2)$[4]. See fig. 4.

In case that M_1 and M_2 do not share any object-type between them (i.e. two disjoint sets of fact-types), the composition ($M_1 \oplus M_2$) is considered an incompatible operation[5], as there is no model that can satisfy both M_1 and M_2. Notice that in case an object-type is specified as "lexical" in one module and as "non-lexical" in another (e.g. 'Account'), then in the composition, this object-type is considered "non-lexical".

Step 2: Combining constraints

When composing two modules, the *combination* of all constraints ($\Omega_1 \oplus \Omega_2$) should be syntactically valid according to the ORM syntax. For example, some constraints need to be syntactically combined into one constraint. *The combination of a set of constraints should imply all of them*. Furthermore, some logical (i.e. satisfiability and implication) validations are also performed in this step, e.g. in case of combining two constraints that contradict or imply each other. In the following, we show how all ORM constraints can be combined.

Step 2.1: Combining value constraints

Given two value constraints $T.v_1$ and $T.v_2$ on the same object-type T, (notice that v_1 and v_2 are two sets of values), their combination is the intersection $T.v = T.v_1 \cap T.v_2$, see fig. 5(a). If $T.v_1 \cap T.v_2$ is empty, then the composition ($M_1 \oplus M_2$) is considered as incompatible operation, because the value constraints contradict each other and thus the object type cannot be satisfied, see fig. 5(b).

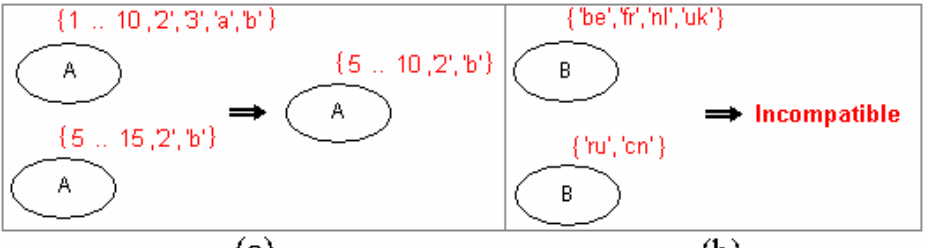


Fig. 5. Combining value constraints

Step 2.2: Combining mandatory constraints

When composing two modules, all mandatory constraints are included in the composition without any specific combining operation.

Step 2.3: Combining disjunctive mandatory

When composing two modules, all disjunctive mandatory constraints are included in the composition without any specific combining operation.

[4] T refers to a Term (concept label), r refers to a role, r' refers to an inverse role.

[5] In practice, we weaken this requirement to allow the composition of disjoint modules. For example, in case one wishes to compose two disjoint modules and later compose them within a third module that results in a joint composition.

Step 2.4: Combining uniqueness and frequency constraints
When composing modules, uniqueness and frequency constraints are combined as follows:

- As internal uniqueness implies predicate uniqueness [H89], the combination of these two constraints is internal uniqueness (see fig. 6. (a) and (b)).
- In case of internal uniqueness and frequency constraints on the same role (see fig. 6(c)), the composition of ($M_1 \oplus M_2$) is considered an **incompatible operation**, because the two constraints contradict each other [H89], and thus the role cannot be satisfied. Recall that a frequency of maximum 1 is considered internally uniqueness (see fig. 6(d)).
- In case of two frequency constraints on the same role, FC_1(min-max) and FC_2(min-max), the combination FC(min-max) is calculated as FC.min = MaxOf(FC_1.min, FC_2.min) and FC.max = MinOf(FC_1.max, FC_2.max), see fig. 6(e). In case the FC.min > FC.max, see fig. 6(f), then the composition of ($M_1 \oplus M_2$) is considered an **incompatible operation**, because the two constraints are in conflict each other, and the role cannot be satisfied.
- In other cases, all constraints are included in the composition without any specific combining operation.

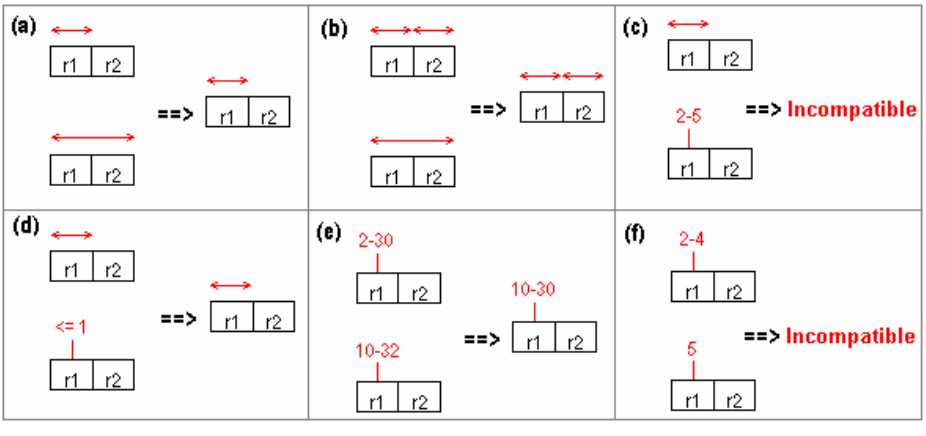


Fig. 6. An example of combining uniqueness and frequency constraints

Step 2.5: Combining set-comparison constraints
Combining set-comparison constraints across two modules is performed in the following steps:

- Each exclusion constraint that spans more than two singles or sequences of roles (called "multiple" exclusion) is converted into pairs of exclusions[6], such in Fig. 7.

[6] This conversion is temporary for reasoning purposes, so it will not appear in the final result of the composition. Notice that "a single exclusion constraint a cross *n* roles replaces *n*(*n*-1)/2 separate exclusion constraints between two roles" [H01].

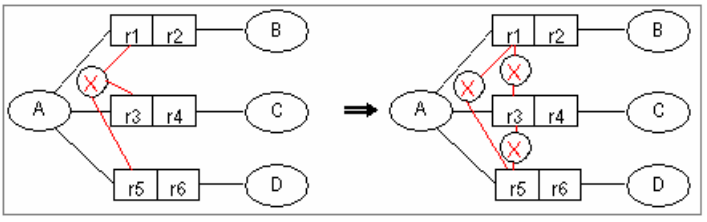


Fig. 7. Converting multiple exclusions into pairs of exclusions

- When combining a subset (or equality) in one module and an exclusion in another, the composition of ($M_1 \oplus M_2$) is considered an **incompatible operation**, because the two constraints contradict each other, and so both roles cannot be satisfied. See fig. 8.
- As equality implies subset (but not vice versa) [H89], when combining a subset in one module and equality in another module, or when combining two subset constraints that are opposite to each other, the combination is always equality. See Fig. 9.

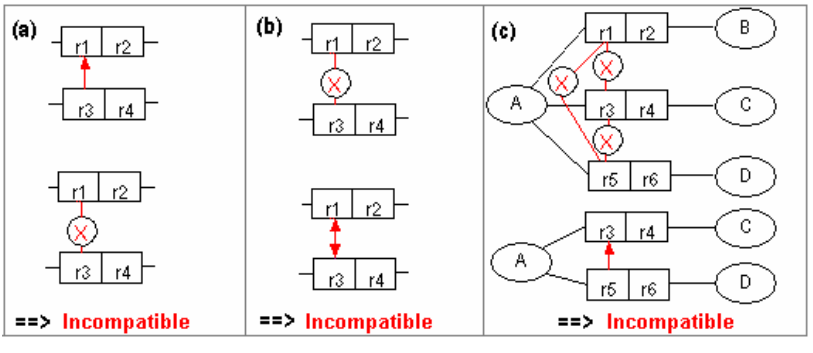


Fig. 8. Combining subset (or equality) with exclusion

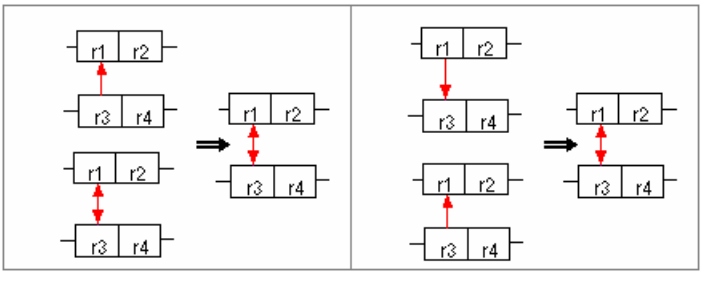


Fig. 9. Combining subset and equality constraints

Step 2.6: Combining subtype constraints (total, exclusive)

When composing two modules, all subtype constraints are included in the composition without any specific combining operation.

Step 2.7: Combining ring constraints

ORM allows ring constraints to be applied to a pair of roles that are connected directly to the same object-type in a fact-type, or indirectly via supertypes. Six types of ring constraints are supported by ORM: antisymmetric (ans), asymmetric (as), acyclic (ac), irreflexive (ir), intransitive (it), and symmetric (sym) [H01][H99]. The relationships between the six ring constraints are formalized by [H01] using the Eular diagram as in fig. 10. This formalization helps one to visualize the implication and incompatibility between the constraints. For example, one can see that acyclic implies reflexivity, intransitivity implies reflexivity, the combination between antiasymmetric and reflexivity is exactly asymmetric, and acyclic and symmetric are incompatible.

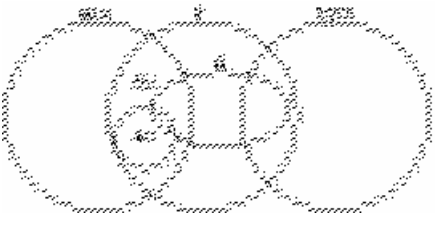


Fig. 10. Relationships between ring constraints [H01]

When composing two modules, ring constraints are combined based on the formalization in fig. 10. Any combination of ring constraints should be compatible, i.e. there is an intersection between their zones in the Eular diagram, e.g. see fig. 11 (a). Otherwise, the composition of $(M_1 \oplus M_2)$ is considered an **incompatible operation**, because the combined rings constraints conflict each other, and thus the role cannot be satisfied. See fig. 11 (b).

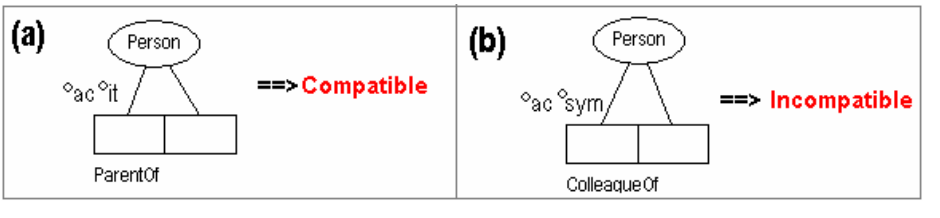


Fig. 11. Examples of compositions ring constraints

4 Discussion, Conclusions and Future Work

This paper has presented an approach to modularize and automatically compose ORM schemes. This approach is fully implemented in DogmaModeler [J05], which is a

software tool for modeling ontologies and business rules using the ORM graphical notation. DogmaModeler enables users to create, compose, add, delete, manage, and browse ORM (modular) schemes. DogmaModeler also implements a library of ORM modular schemes, allowing different metadata standards (e.g. Dublin-Core, LOM, etc.) to be used for describing modules. This approach has been also used in a real-life case study (CCFORM EU project, IST-2001-34908, 5th framework.) for developing modular axiomatizations of costumer complaints knowledge, see [J05][JVM03] for the experience and lessons learned.

Although we assume in our formal framework (in section 2) that the composition is terminated in case of unsatisfiability, it is not necessary for the resultant composition, in our algorithm of composing ORM schemes (in section 3) to be satisfiable, thus our algorithm is called *incomplete*. This is because the general problem of determining consistency for all possible constraint patterns in ORM is undecidable [H97]. A complete semantic tableaux algorithm for deciding the satisfiability of ORM schemes (a research topic by itself) is not a goal of this paper. See our pattern-based approach in [JH05] for reasoning about the satisfiability of ORM schemes.

As an upcoming effort, we plan to map ORM into the ***DLR*** Description Logic [CDLNR98], which is a powerful and decidable fragment of first order logic. In this way, the satisfiability of ORM schemes can be completely verified, and so our algorithm can be called complete. Furthermore, this will allow us to reuse our approach to modularize and compose ***DLR*** knowledge bases.

Acknowledgement. We are in debt to Robert Meersman, Stijn Heymans, Olga De Troyer and Andriy Lisovoy for their comments, discussion, and suggestions on the earlier version of this work.

References

[BBDD97] Briand, L.C., Bunse, C., Daly, J.W. and Differding, C.: An Experimental Comparison of the Maintainability of Object-Oriented and Structured Design Documents. In: Empirical Software Engineering, Vol. 2, No. 3. (1997) pp. 291–312.

[BGH99] Bird, L., Goodchild, A., Halpin, T.A.: Object Role Modelling and XML-Schema. In: Laender, A., Liddle, S., Storey, V. (eds.): Proceedings of the 19th International Conference on Conceptual Modeling (ER'00). LNCS, Springer Verlag (1999)

[BHW91] van Bommel, P., ter Hofstede, A.H.M. , van der Weide, Th.P. : Semantics and verification of object role models. Information Systems, 16(5). October (1991) 471–495

[BS03] Borgida A., Serafini L.: Distributed Description Logics: Assimilating Information from Peer Sources. In: Aberer K., March S., and Spaccapietra S., (eds.): Journal on Data Semantics, Vol. 2800. LNCS, Springer, ISBN: 3-540-20407-5. October (2003) pp. 153–184

[CDLNR98] Calvanese, D., De Giacomo, G., Lenzerini, M., Nardi, D., Rosati, R.: Information integration: Conceptual modeling and reasoning support. In Proceedings Of the 6th International Conference on Cooperative Information Systems (CoopIS'98). (1998) pp. 280-291

[DJM02a] Demey, J., Jarrar, M., Meersman, R.: A Conceptual Markup Language that supports interoperability between Business Rule modeling systems. Proceedings of the Tenth International Conference on Cooperative Information Systems (CoopIS 02). Springer Verlag LNCS 2519. (2002) pp. 19–35

[H01] Halpin, T.: Information Modeling and Relational Databases. 3rd edn. Morgan-Kaufmann. (2001)

[H04] Halpin, T.: Business Rule Verbalization. In Doroshenko, A., Halpin, T., Liddle, S., Mayr H. (eds): Information Systems Technology and its Applications, 3rd International Conference (ISTA'2004), LNI 48 GI ISBN 3-88579-377-6, (2004) pp:39-52.

[H89] Halpin, T.: A logical analysis of information systems: static aspects of the data-oriented perspective. PhD thesis, University of Queensland, Brisbane. Australia. (1989)

[H97] Halpin, T.: An Interview- Modeling for Data and Business Rules. In: Ross, R. (eds.): Database Newsletter. vol. 25, no. 5. (Sep/Oct 1997). -This newsletter has since been renamed Business Rules Journal and is published by Business Rules Solutions, Inc.

[H99] Halpin, T.: UML data models from an ORM perspective: Part 7. Journal of Conceptual Modeling. InConcept. February (1999)

[J05] M. Jarrar. Towards Methodological Principles for Ontology Engineering. PhD thesis, Vrije Universiteit Brussel, 2005.

[JDM03] Jarrar M., Demy J., Meersman R.: On Using Conceptual Data Modeling for Ontology Engineering. In: Aberer K., March S., and Spaccapietra S., (eds.): Journal on Data Semantics, Special issue on "Best papers from the ER/ODBASE/COOPIS 2002 Conferences", LNCS Vol. 2800, Springer. ISBN: 3-540-20407-5. October (2003) pp. 185–207

[JH05] Jarrar, M., Heymans, S.: Unsatisfiability Reasoning in ORM Conceptual Schemes. Technical Report, Vrije Universiteit Brussel, 2005.

[JM02b] Jarrar, M., Meersman, R.: Scalability and Knowledge Reusability in Ontology Modeling. Proceedings of the International conference on Infrastructure for e-Business, e-Education, e-Science, and e-Medicine (SSGRR'2002s) (2002)

[JVM03] Jarrar, M., Verlinden, R., Meersman, R.: Ontology-based Customer Complaint Management. In: Jarrar M., Salaun A., (eds.): Proceedings of the workshop on regulatory ontologies and the modeling of complaint regulations, Catania, Sicily, Italy. Springer Verlag LNCS. Vol. 2889. November (2003) pp. 594–606

[N99] North, K.: Modeling, Data Semantics, and Natural Language. In: New Architect magazine (1999)

[P72] Parnas, D. L.: On the criteria to be used in decomposing system into modules. Communications of the ACM, Vol. 15, No. 12. December (1972) pp. 1053–1058

[S85] Shoval, P.: Essential information structure diagrams and database schema design. Information Systems, 10(4). (1985) pp. 417-423

[SK03] Stuckenschmidt H., Klein M.: Modularization of Ontologies -WonderWeb: Ontology Infrastructure for the Semantic Web. Deliverable 21. WonderWeb Project (IST 2001-33052) (2003)

[SMS+05] Spaccapietra, S., Menken, M., Stuckenschmidt, H., Wache, H., Serafini, L., Tamilin, A., Jarrar, M., Porto, F., Parent, C., Rector, A., Pan, J., D'Aquin, M., Lieber, J., Napoli, A., Stoilos, G., Tzouvaras, V., Stamou, G.: Report on Modularization of Ontologies. Deliverable D2.1.3.1 (WP2.1), KnowledgeWeb project. EU-IST Network of Excellence (NoE) IST-2004-507482 (2005)

[SP94] Spaccapietra, S., Parent, C.: View Integration: A Step Forward in Solving Structural Conflicts. IEEE Transactions on Data and Knowledge Engineering 6(2). (1994)

[SWCH01] Sullivan, k., William, G., Cai, Y., Hallen, B.: The structure and value of modularity in software design. Journal SIGSOFT Software Engineering Notes. Vol. 26, number 5. ACM Press. Issn: 0163-5948. (2001) pp. 99–108

[V83] Vermeir D.: Semantic Hierarchies and Abstraction in Conceptual Schemata. Journal of Information Systems. Vol. 8, No. 2. (1983) pp. 117–124

[VB82] Verheijen, G., van Bekkum, P.: NIAM, aN Information Analysis Method. In: Olle, T.W., Sol, H., Verrijn-Stuart, A. (eds.), IFIP Conference on Comparative Review of Information Systems Methodologies, North-Holland. (1982) pp. 537–590

Modelling Context Information with ORM*

Karen Henricksen[1], Jadwiga Indulska[2], and Ted McFadden[1]

[1] CRC for Enterprise Distributed Systems Technology (DSTC)
karen@itee.uq.edu.au, mcfadden@dstc.edu.au
[2] School of Information Technology and Electrical Engineering,
The University of Queensland
jaga@itee.uq.edu.au

Abstract. Context-aware applications rely on implicit forms of input, such as sensor-derived data, in order to reduce the need for explicit input from users. They are especially relevant for mobile and pervasive computing environments, in which user attention is at a premium. To support the development of context-aware applications, techniques for modelling context information are required. These must address a unique combination of requirements, including the ability to model information supplied by both sensors and people, to represent imperfect information, and to capture context histories. As the field of context-aware computing is relatively new, mature solutions for context modelling do not exist, and researchers rely on information modelling solutions developed for other purposes. In our research, we have been using a variant of Object-Role Modeling (ORM) to model context. In this paper, we reflect on our experiences and outline some research challenges in this area.

1 Introduction to Context Modelling

Context-awareness has recently emerged as a popular approach for building applications for mobile and pervasive computing environments that are capable of automatically adapting to their environments and reducing the need for explicit directions from the user. Context-aware applications monitor and respond to information about the context of use, such as the available computing resources and the current location and activity of the user. Context-aware applications typically obtain this information - which is termed *context information* - from varied sources, including people, sensors (e.g., GPS receivers, tilt sensors and accelerometers), network monitors, and other context-aware applications. Common examples of context-aware applications include tourist guides that present information that is tailored according to the user's location and preferences, and mobile phones that adapt their ringing behaviour depending on where the user currently is, who they are with, and what they are likely to be doing. A variety

* The work reported in this paper has been funded in part by the Co-operative Research Centre for Enterprise Distributed Systems Technology (DSTC) through the Australian Federal Government's CRC Programme (Department of Education, Science, and Training).

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 626–635, 2005.
© Springer-Verlag Berlin Heidelberg 2005

of emerging 'intelligent environments' - such as homes, hospitals and meeting rooms that are able to sense the activities of their occupants and leverage this information to automatically derive those occupants' requirements - can also be considered context-aware.

Early context-aware applications, which typically relied on only one or two simple types of sensed context information, were generally constructed by writing software modules to directly query sensors and carry out simple types of interpretation of the sensor outputs. In these applications, the model of context information - or, more specifically, the *kinds* of context used, and their *representations* or *formats* - were fixed and closely intertwined with the application logic. Today, the shortcomings of this approach are widely acknowledged. Increasingly, developers of context-aware applications are seeking techniques for modelling context information in a uniform way, in order to:

- promote sharing of context information and context-sensing infrastructure between applications;
- facilitate complex queries over multiple types of context information;
- decouple the details of the model from the application logic, thereby simplifying the process of evolving the model; and
- provide a common framework for representing both sensor-derived data and information from other sources, both of which are important for context-aware applications.

As yet, there are no well-established techniques for modelling context, as Strang and Linnhoff-Popien [1] demonstrated in their recent survey. Most researchers have adopted modelling approaches from other fields, such as database modelling (ORM, ER) and the Semantic Web (CC/PP, DAML+OIL, OWL). A major advantage of these approaches is that most are widely understood and supported by common tools and query languages. On the other hand, there is often a considerable degree of mismatch between the capabilities of these modelling approaches and the characteristics of the context information that needs to be modelled, which necessitates extensions or work-arounds. It is this problem that we address in this paper, drawing on our experiences with using an ORM-based context modelling approach to develop a variety of context-aware applications. We highlight some requirements for modelling context that are not met by ORM (or other similar modelling approaches) and present a set of novel extensions that we have developed to address some of these requirements. We also show how our extended variant of ORM is used to support the development of context-aware applications, and outline some of the remaining challenges in the area of context modelling. We assume that the reader already has a basic familiarity with ORM; for a comprehensive treatment, we refer the reader to the excellent book by Halpin [2].

2 Requirements for Modelling Context Information

The traditional use of ORM is the modelling of business domains, in order to support the development of databases that are principally populated and used by

humans. Our goals in modelling context information are quite different. Context models are mapped to information repositories that are populated by many different entities, including humans, hardware and software sensors and context-aware applications. Context repositories are primarily queried by context-aware applications, which use the information to determine the current situation and requirements of their users. These differences in information production and consumption mean that ORM does not provide the most natural solution for modelling context. In this section, we highlight a set of specific context modelling problems which ORM either does not address well or at all.

2.1 Distinguishing Information from Different Sources

In order to manage and use context information effectively, it is necessary to distinguish between sensed, human-supplied and application-supplied information. Sensed context information is generally updated frequently but can be inaccurate due to problems like noise, calibration/configuration errors, and so on. In comparison, user-supplied information is updated infrequently. It is typically accurate initially, but becomes less reliable over time. Application-supplied information has characteristics that fall between those of sensed and user-supplied information. We require techniques for associating fact types with information sources, and, in some cases, annotating individual facts with specific metadata about their sources (e.g., sensor or user IDs).

2.2 Allowing Inconsistency and Incompleteness

Conflicting information is a common problem in context-aware systems; different sensors, for example, may report different values, and it may not be possible to determine which value is the correct one. Likewise, incomplete information is the norm rather than the exception. ORM uses a variety of constraints, such as uniqueness constraints and mandatory role constraints, to prevent such problems from arising; these are mapped to database constraints, so that updates that violate the constraints are rejected at run-time. This is acceptable when information is inserted by humans who can investigate the cause of the conflicts and resolve them. However, it is not appropriate for context-aware systems. Instead, inconsistencies and incompleteness should be allowed in a controlled fashion, and appropriately handled by context-aware applications. This requires separate modelling constructs to capture true domain constraints, which match the real world semantics, and the looser integrity constraints that should be enforced by context repositories.

2.3 Modelling Temporal Data and Constraints

Context-aware applications are frequently not only interested solely in the current context, but also in future or past states, or changes in state over time. Therefore, it is often desirable to model *histories* of certain types of context information. Although histories can be modelled in ORM by explicitly including time as an additional role in fact types, this solution is clumsy; it is more

natural to provide dedicated modelling constructs for temporal fact types, including special temporal constraint types. A considerable amount of work has been done in the area of modelling temporal data, and some solutions do exist for extending ORM with temporal concepts [3]; however, these are not yet part of the standard ORM notation.

2.4 Modelling Information Quality

As discussed in Sections 2.1 and 2.2, context information is often imperfect, and context-aware applications must be capable of operating under this assumption. To allow applications to make informed decisions about which context information should be trusted and which should not, quality metadata is required. Appropriate types of metadata vary between fact types; for example, a fact describing a person's current activity might be associated with a timestamp, while a fact describing a person's current location coordinates might also be annotated with the standard error associated with the sensor supplying the information. Although quality could be modelled simply by including additional roles in fact types, this solution becomes undesirable when it comes to reasoning about context information, as quality values cannot be distinguished from ordinary values in facts.

2.5 Modelling Information Ownership

A final requirement is imposed by privacy requirements in context-aware systems. In business domains, an information repository generally falls under the control of a single business entity, which can set global access control policies for users of the repository; unfortunately, this is not the case for context repositories, which may combine sensitive information belonging to many users [4]. This necessitates a more complex model of ownership and control. One way to address this requirement is to extend the context model with statements that explicitly distribute the ownership of individual information types (i.e., fact types) amongst a set of people and/or other entities.

3 The Context Modelling Language

Although solutions exist for some of the problems discussed in the previous section (for example, for modelling temporal data [5] and information quality [6]), there is no single modelling approach that addresses all of the problems in a cohesive way. In this section, we outline a set of extensions to ORM that we developed to address most of the problems. For want of a better name, we refer to this ORM variant as the Context Modelling Language (CML).

3.1 Source Annotations

Our first extension allows fact types to be characterised in terms of the persistence and source of the information that they capture. The motivation for this

extension was provided in Section 2.1. First, we differentiate between *static* and *dynamic* fact types. Static fact types represent invariant properties of a context-aware system. Static facts are very simple to manage; they are usually stored indefinitely in context repositories for querying by context-aware applications.

Dynamic fact types are classified according to source. *Sensed* fact types represent information supplied by hardware or software sensors, while *profiled* fact types represent information supplied by users or context-aware applications. The former are generally frequently updated, while the latter are not; therefore, they suit different styles of management within context repositories. Sensed information that is only infrequently queried may not be kept up to date within context repositories (so as to conserve resources), but instead loaded on demand by querying the appropriate sensors.

The annotations we use to represent static, profiled and sensed fact types are illustrated in Fig. 1 (a)-(c). Note that these annotations are never attached to ORM's derived fact types.

3.2 Alternative Fact Types

In order to deal with conflicts of the kind that we described in Section 2.2, in which sensors report different values for some type of context, we introduce the notion of *alternatives*. Alternatives are mutually exclusive facts (of the same fact type) that have been reported about some entity or entities. Example alternatives are "Michelle is located in Sydney" and "Michelle is located in New York".

To selectively allow alternatives, we introduce an alternative fact type. This is annotated with an '*a*' symbol and a special alternative uniqueness constraint, as shown in Fig. 1 (d). The uniqueness constraint always spans n-1 roles, where n is the arity of the fact type. The role *not* spanned by the constraint is known as the *alternative role*. Alternative uniqueness constraints are distinguished from ordinary uniqueness constraints to indicate their distinct semantics. An alternative uniqueness constraint is a domain constraint that is not strictly enforced by context repositories; instead the constraint is effectively extended over the alternative role to allow alternative values for this role. Alternative facts have different semantics to ordinary facts, which needs to be taken into account when querying context repositories. We discuss this issue briefly in Section 3.6.

3.3 Temporal Fact Types

To accommodate histories of context information, we extended ORM with a temporal fact type. This extension uses an entirely different notation to the recent TORM proposal [3], as it pre-dates TORM. The notation is shown in Fig. 1 (e). A fact type is marked as a temporal fact type using the '[ ]' annotation, which has the effect of associating all facts with a *valid time* [7], expressed as an interval having a start time and an end time.

Uniqueness constraints on temporal fact types can be either *snapshot* or *lifetime* constraints, in the terminology of [8]. A single fact type may have both types of uniqueness constraint. CML adopts the convention that all constraints on temporal fact types are, by default, lifetime constraints. This preserves the normal

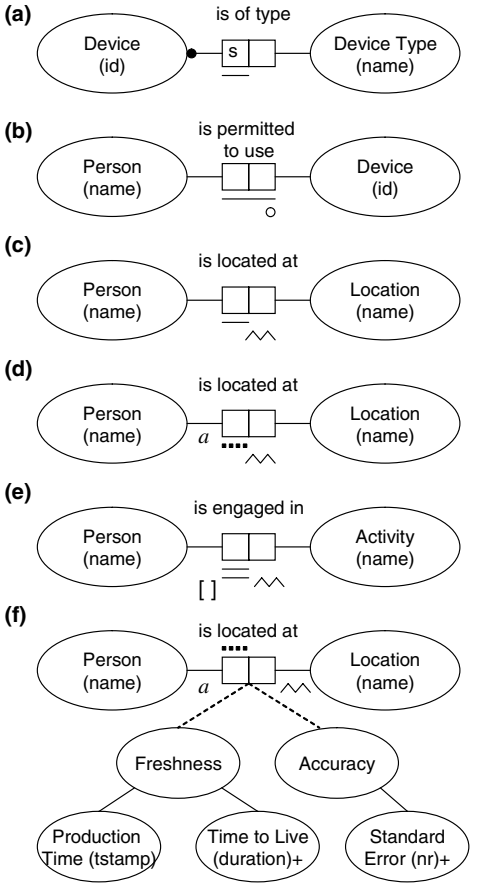


Fig. 1. CML's context modelling extensions. **(a)**, **(b)** and **(c)** show static, profiled and sensed fact types, respectively. **(d)** provides an example of an alternative fact type; this allows multiple location readings to be associated with each person. **(e)** illustrates a temporal fact type used to capture histories of user activities. **(f)** shows an alternative fact type with quality annotations.

semantics of the constraints when the timestamps that are implicit in temporal fact types are regarded as objects participating in ordinary roles. Snapshot uniqueness constraints are drawn using the special notation shown in Fig. 1(e). These express constraints that apply at any given point in time, but not globally over fact histories. For example, the constraint in Fig. 1 (e) indicates that a person has at most one activity at any time (but can have many activities within a history). An external variant is also supported, in which double, rather than single, lines are used to connect the encircled u to the participating roles.

Care must be taken when combining temporal fact types with other kinds of fact types. In general, derived temporal fact types are nonsensical unless at

least one of the base fact types involved in the derivation is also temporal. There are also some complications associated with combining temporal and alternative fact types, which are outside our scope here, but are documented in [9].

3.4 Quality Annotations

To support decision making about imperfect context information by context-aware applications, CML provides constructs for annotating fact types with quality annotations that effectively allow quality metadata to be attached to individual facts. CML's quality constructs are partially inspired by work of Wang et al. [6,10] that addressed quality modelling in relation to ER. As illustrated in Fig. 1 (f), each fact type can be annotated with zero or more quality parameters (here, *freshness* and *accuracy*). These parameters are associated with one or more metrics (*production time*, *time to live* and *standard error*), which indicate how the quality is measured/recorded for each fact.

An alternative modelling approach, which does not require special constructs, would be to objectify the fact type to which the quality annotations are attached and add one new binary fact type for each quality metric, in which the objectified fact type plays one role and the metric plays the other. However, our notation is more natural and compact, and can have its own specialised mapping when creating context repositories from CML models.

3.5 Ownership Statements

Finally, to support privacy, we have created a textual notation for specifying the ownership of facts. Ownership statements are specified as part of a *context schema*, which is a textual form of a CML model that we created as a convenient form of input for tools that perform manipulations and mappings on context models. We will discuss these tools in Section 4.

To keep the task of specifying ownership manageable even for large numbers of fact types, we primarily associate ownership with objects rather than facts, and then assign a default ownership to each fact by forming the union of the ownerships of the objects participating in the fact. This default ownership can also be explicitly overridden by providing ownership statements for fact types. A full discussion of the ownership scheme can be found in an earlier paper [4].

3.6 Relational Mapping, Querying and Interpretation

One of the attractions of ORM is its mapping to the relational model, and we have extended this mapping to incorporate our context modelling constructs. It is not possible, owing to space constraints, to describe the details of the mapping procedure in this paper; however [9] provides a full discussion. In Section 4, we will briefly describe a tool that we have developed to automate the procedure.

While there are no *representational* problems associated with mapping our extended ORM to the relational model, there are problems of *interpretation*. As mentioned in Section 3.2, an alternative fact does not have the same semantics as an ordinary fact. To overcome this problem, we provide our own query layer

for context repositories that maps portions of context queries to standard relational database queries expressed in SQL. The query layer provides evaluation of context information using a three-valued logic which is able to accommodate alternative facts. We plan to extend the query mechanism to provide more sophisticated treatment of quality annotations, unknowns and temporal facts.

4 Tool Support for Developing Context-Aware Systems

We have developed a set of tools and infrastructural components to support software engineers in the task of constructing and deploying context-aware applications that use CML models. These include a context management layer that augments a relational database with additional functionality required for storing and querying CML models, and a schema compiler toolset that supports a variety of transformations on CML models. The schema compiler tools accept a context model description in the context schema notation we discussed in Section 3.5, perform checks to verify the integrity of the model, and then produce one or more of the following outputs:

- SQL scripts to load and remove context model definitions from relational databases;
- model-specific helper classes, for Java and Python, that can be used by application developers to simplify source code concerned with context queries and updates; and
- context model interface definitions that can be compiled to stubs that can be used by applications to easily transmit or receive context information without dealing with any of the protocols used for remote communication.

The tools and infrastructure are documented further in [11] and [12].

5 Open Research Problems

Context modelling has recently become a hot topic in the field of pervasive computing, and numerous modelling approaches have appeared since we first began working on CML in 2002. However, many open research problems remain. In particular, more work is needed in relation to modelling, querying and reasoning over imperfect context information, in order to adequately address problems such as sensing errors and sensor failures. Although solutions exist in other fields for dealing with imperfect information, it is likely that no solution will provide a perfect match with the requirements of context-aware systems.

Further work is also needed to develop sophisticated context management systems, which must be radically different to the average relational database system. One important issue is traceability - that is, being able to track context information from its source, through various forms of processing (e.g., sensor fusion), to the repositories in which the information eventually resides. This kind of tracking is needed to link incorrect context information to failed or misconfigured components, thereby enabling debugging and repair. Context management

systems must also address issues of scalability, performance and distribution in order to satisfy the requirements of pervasive systems, which may involve very large number of mobile sensors, applications and users. So far, only small prototypes have been developed which have not needed to address these problems.

Finally, work is needed on how to provide interoperability between multiple context-aware systems that each possess their own context models, and possibly also their own context modelling approaches. Some early work has already begun in this area using ontology standards such as DAML+OIL and OWL [13,14].

6 Concluding Remarks

In this paper, we introduced CML, an ORM-based approach for modelling the context information required by context-aware applications. To date, we have used CML to produce context models for several context-aware communication applications [15,16], a vertical handover application [11], and applications to support independent living of elderly people living in 'smart homes' [17]. With the exception of the independent living applications, all of these applications have been fully implemented. In conjunction with these efforts, we have built a suite of tools to support software engineers in building and deploying applications that use CML models, leveraging the mapping to the relational model.

Although numerous research challenges remain in relation to context modelling, we believe that CML is one of the most viable of the currently available solutions, and is well positioned to serve as a platform for investigating new modelling constructs that will begin to address these challenges. We have found ORM's fact types to provide a natural basis for extension and annotation with metadata and constraints - more so than attribute-based modelling approaches such as ER and UML, and ontology languages, such as OWL, which we have also evaluated as techniques for context modelling, as discussed in [18] and [14].

References

1. Strang, T., Linnhoff-Popien, C.: A context modeling survey. In: UbiComp 1st International Workshop on Advanced Context Modelling, Reasoning and Management, Nottingham (2004) 34–41
2. Halpin, T.A.: Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design. Morgan Kaufman, San Francisco (2001)
3. Pornphol, P., Chittayasothorn, S.: A temporal relational and object relational database design technique. In: SoutheastCon. (2004) 54– 59
4. Henricksen, K., Wishart, R., McFadden, T., Indulska, J.: Extending context models for privacy in pervasive computing environments. In: 2nd International Workshop on Context Modelling and Reasoning (CoMoRea), PerCom'05 Workshop Proceedings, IEEE Computer Society (2005) 20–24
5. Gregersen, H., Jensen, C.S.: Temporal entity-relationship models - a survey. IEEE Transactions on Knowledge and Data Engineering **11** (1999) 464–497
6. Wang, R., Reddy, M.P., Kon, H.: Towards quality data: An attribute-based approach. Decision Support Systems **13** (1995) 349–372

7. Jensen, C.S., et al.: The consensus glossary of temporal database concepts - February 1998 version. In: Temporal Databases: Research and Practice. Volume 1399 of Lecture Notes in Computer Science., Springer (1998) 367–405
8. Tauzovich, B.: Towards temporal extensions to the entity-relationship. In: 10th International Conference on the Entity-Relationship Approach (ER), San Mateo (1991) 163–179
9. Henricksen, K.: A Framework for Context-Aware Pervasive Computing Applications. PhD thesis, School of Information Technology and Electrical Engineering, The University of Queensland (2003)
10. Storey, V., Wang, R.: Modeling quality requirements in conceptual database design. In: 3rd Conference on Information Quality (IQ), Cambridge (1998) 64–87
11. Henricksen, K., Indulska, J., McFadden, T., Balasubramaniam, S.: Middleware for distributed context-aware systems. International Symposium on Distributed Objects and Applications (DOA) (to appear) (2005)
12. McFadden, T., Henricksen, K., Indulska, J.: Automating context-aware application development. In: UbiComp 1st International Workshop on Advanced Context Modelling, Reasoning and Management, Nottingham (2004) 90–95
13. Strang, T., Linnhoff-Popien, C., Frank, K.: CoOL: A Context Ontology Language to Enable Contextual Interoperability. In: 4th International Conference on Distributed Applications and Interoperable Systems (DAIS). Volume 2893 of Lecture Notes in Computer Science., Springer (2003) 236–247
14. Henricksen, K., Livingstone, S., Indulska, J.: Towards a hybrid approach to context modelling, reasoning and interoperation. In: UbiComp 1st International Workshop on Advanced Context Modelling, Reasoning and Management, Nottingham (2004) 54–61
15. Henricksen, K., Indulska, J.: A software engineering framework for context-aware pervasive computing. In: 2nd IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE Computer Society (2004) 77–86
16. McFadden, T., Henricksen, K., Indulska, J., Mascaro, P.: Applying a disciplined approach to the development of a context-aware communication application. In: 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE Computer Society (2005) 300–306
17. Indulska, J., Henricksen, K., McFadden, T., Mascaro, P.: Towards a common context model for virtual community applications. In: 2nd International Conference on Smart Homes and Health Telematics (ICOST). Volume 14 of Assistive Technology Research Series., IOS Press (2004) 154–161
18. Henricksen, K., Indulska, J., Rakotonirainy, A.: Modeling context information in pervasive computing systems. In: 1st International Conference on Pervasive Computing (Pervasive). Volume 2414 of Lecture Notes in Computer Science., Springer (2002) 167–180

Using Object Role Modeling for Effective In-House Decision Support Systems

Eric John Pierson and Necito dela Cruz

4100 Hamline Ave N, St. Paul, MN USA 55112
Eric.Pierson@Guidant.com
Necito.delaCruz@Guidant.com

Abstract. This is a practical application article that illustrates how Guidant Corporation, a medical device manufacturer of cardiac rhythm management (CRM) devices, utilizes Object Role Modeling (ORM). While some business environments allow only lip service to be paid to best practices, the cardiac rhythm management industry does not have room for error. These medical devices control the heart – lives depend on them. This article discusses Guidant's use of ORM as a best practice to document the business data rules and establish them as the "single point of truth" across the spectrum of decision support system (DSS) activities.

1 Introduction

This article illustrates what Object Role Modeling means to the challenge of advancing the state of the art of business at Guidant. Guidant and many other companies outside of the medical device industry are rapidly developing in-house decision support solutions. Decision support tools such as Cognos' ReportNet (used at Guidant for these tasks) are making it easier for companies to manage their own success and significantly reduce their dependencies on external specialty vendors. The take-away from this article is that decision support tools are evolving rapidly, and that companies who purchase these tools will need practitioners of ORM to provide best practice leadership in order to succeed.

2 What It Takes to Create Guidant DSS Solutions

Recent advances in DSS technology such as Cognos' ReportNet suite of tools are breaking down barriers and making it easier for companies to build their own decision support solutions in-house. With ReportNet, Guidant is preserving the integrity of intellectual assets and reducing dependencies on external vendors. ReportNet assists the Information Management departments to engage their business customers to take an active part in creating, supporting, and growing highly customized decision support systems. This is in direct contrast to more traditional approaches to internally developed decision support systems where business units were limited to being data consumers and were restricted to making use of the data via ad-hoc and standard reports that had been pre-designed on their behalf. The ability for Guidant business groups to use ReportNet to make fast changes to manage their own decision support

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 636 – 645, 2005.
© Springer-Verlag Berlin Heidelberg 2005

systems without being fully dependent on their Information Management department allows our company to react to changes faster and with more agility.

As authors with expertise in decision support tools and consulting services, we have a solid understanding of what it takes to successfully implement decision support systems:

1) People with the right skills, accessing the right data through the right tools.
2) Then organizing these people strategically to allow them to drive change and success.

If you think this sounds like a daunting task…you are right, it is!

At Guidant, using Cognos' ReportNet provides many benefits, but as noted above having the right tools is only one piece of our puzzle – skilled people are needed. When DSS specialty vendors create their tools, they have a spectrum of skill sets behind the scenes. The extent of this expertise is rarely visible to their clients – i.e. the clients are not able to see everything involved in order to "pull off" the solution and get the desired results. Their clients are typically focused on the work of using the tools and incorporating them into their business practices – which is a big task in itself.

Assuming that funding, executive buy-in, and decision support strategies are in place, the right skill sets need to be in place when in-housing:

Back-end (getting data from the transactional systems to the DSS)

- Data architect (data modeling and strategy)
- ETL developers (designing, constructing, and managing the flow of data from source systems into the DSS system)
- Business analysts for transactional data (business expertise about data at the source system level)
- Business analysts for analytic data (business expertise about data at the DSS level)
- Database administrator (control and management of servers, environments, and databases)
- Application architects (guiding the design of the environment including security)
- Back-end software experts (experts on the software systems being implemented)
- Senior leadership (sponsorship, funding, and political management)
- Data governance and data stewardship (overall management and regulation of data).

Front-end (The DSS)

- Data architect (DSS tools for presentation layers)
- Account management (managing relationships with business units)
- Business analysis (business use case and business process analysis)
- Front-end software experts (experts on the software systems being implemented and report creators)
- Business solution design (capital committee, departmental on-boarding, expanding and driving deeper use, business-use program development)
- Training/education (tool implementation and business-use training)
- Departmental expertise within user group business departments (business unit experts as first-line help desk)

- Senior leadership sponsors (business-use strategies and return on investment assessment)
- Data governance and data stewardship (overall management and regulation of data).
- Help desk support role and process (user questions and requests).

It is important to keep in mind that a single person may fulfill one or more of these roles. At Guidant we have found that using ORM gives us a clear focus and assists in creating less re-design work. Through this, we have been able to reduce the labor intensity so that while these roles are needed, each role is not always a full-time position.

3 Using ORM for Working Smart

To this point we have discussed Guidant having the right data, the right tools, the right people and skills, and the right organization of those people – what else is missing? Guidant has learned that having a methodology to work smart is a critical ingredient for success. Object Role Modeling provides more for in-house development of DSS than just helping us to generate a solid database – we use ORM to operate as a DSS language that keeps us focused on the truths of our business data.

ORM documents the business data rules which are the established "single point of truth" used by all. The ORM rules are evolved and documented as new truths are discovered. Working smart is working from the same source of truth – regardless of individual roles, skills, or tasks. With this approach to working smart, the truth of the data remains central to the spectrum of DSS activities – as accurate business decisions depend on it.

Historically, the majority of data warehouse initiatives within Guidant overlooked the importance and criticality of understanding data. A common and prevalent mistake was to determine the physical structure of the data at the onset by prematurely adopting the star schema. This often led to inflexible structures that were not as conducive to true ad-hoc environments as was needed.

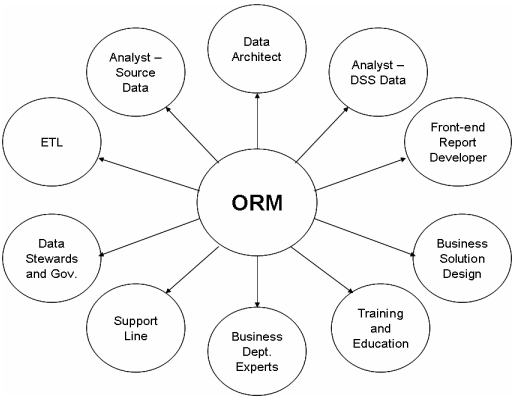
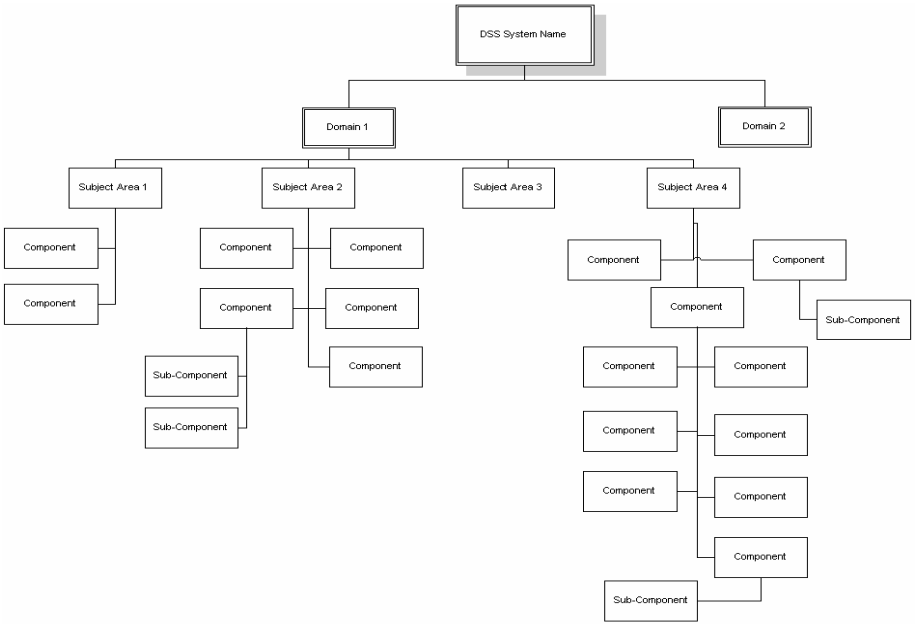


Fig. 1. Illustration of ORM being central across Guidant roles

Guidant's current decision support environments revolve around a more rational approach. We base our work upon the philosophy that for business users to fully drive value from our systems, everyone involved needs to understand the data. Fully understanding the data is the major building block of solid data warehouse architecture.

To better manage and control data analysis activities and to prevent the tendency to get entrenched too deeply in the "weeds" during development work, we utilize the common sense of "divide and conquer." We develop top-down approaches by defining the domains and subject areas based on high level input of our business partners in conjunction with results from structured planning projects and hands-on proof-of-concept prototypes.

A recent implementation of a DSS system for clinical data provides excellent graphic examples of this top-down, divide and conquer approach:


DSS System Name
Domain 1
Domain 2
Subject Area 1
Subject Area 2
Subject Area 3
Subject Area 4
Component
Component
Component

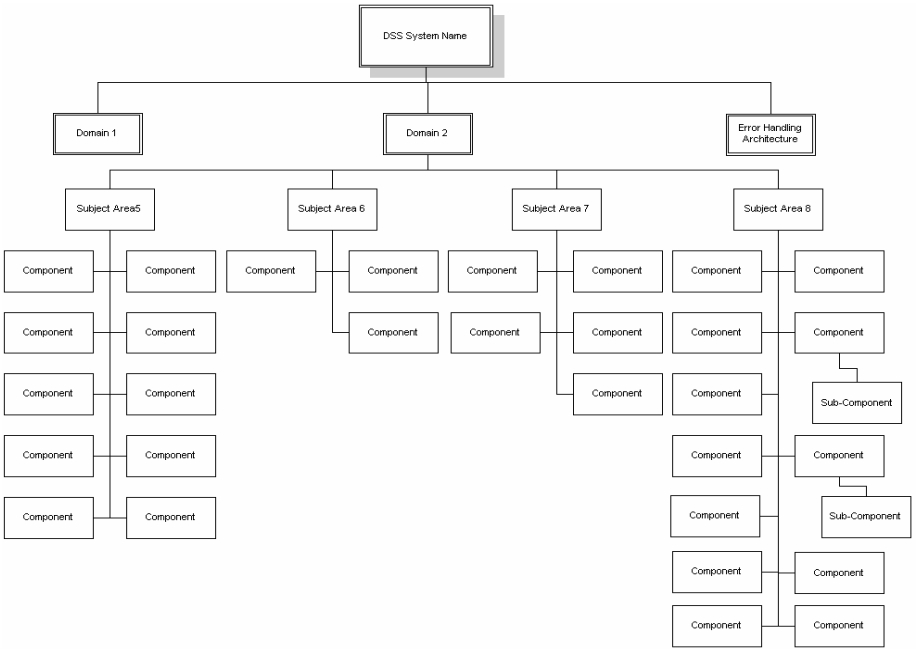


Fig. 3. Second illustration of data decomposition for clinical decision support system. Note that this data decomposition has been de-identified to protect Guidant's intellectual capital.

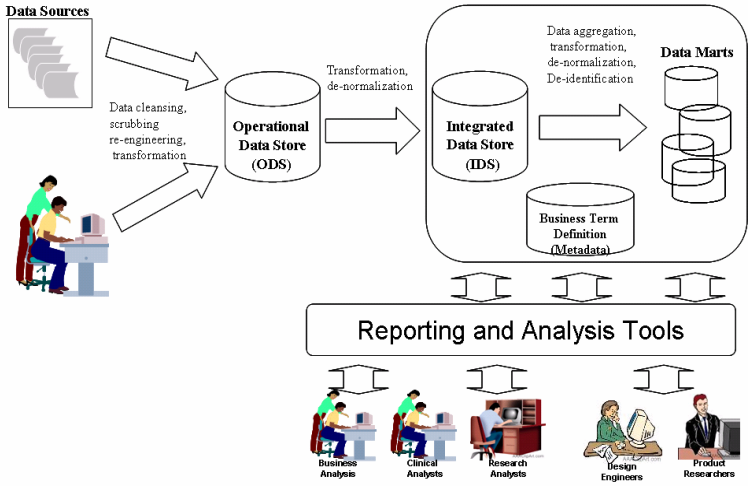


Fig. 4. The Operational Data Store (ODS), the Integrated Data Store (IDS), and the Metadata Store (above) are all ORM-based

In this example, the result of this rigorous data modeling activity is a solid clinical data warehouse architecture that is foundational to strategic decision making at Guidant. ORM conceptual modeling of business facts drove the detailed business data requirements that included:

- Relationships between business objects (data elements);
- Business facts or semantics;
- Business rules and constraints;
- Business terms and vocabularies.

Taking this example a step further, the conceptual business facts gathered and documented through ORM directed the logical and physical structures of this clinical decision support system.

3.1 ORM Benefits Effective In-House Development of Decision Support Systems

There are number of key reasons why Guidant has adopted ORM as a best-practice for designing, implementing, and supporting decision support systems that produce solid results. Consider the following:

ORM Features and Benefits for a Solid Data Warehouse Architecture

Operational Data Store (ODS)	
Features	**Benefits**
Enforces business rules and constraints. Foundation for error checking and correction procedures.	Assures quality and integrity of data. Allows feedback to source system through assuring data validation and improvement to source system. Assists in defining specific data stewardship tasks.
Foundation to data glossary and metadata.	Consistent interpretation of terms and definitions. Contributes to future enterprise-wide data glossary. Acts as an input to user interface layer for data element definition.
Foundation for access and security procedures.	Assists in defining specific data stewardship procedures. Assists in defining data governance policies and procedures.

Foundation for integration of data from different source systems.	Different data source systems will have consistent rules and constraints and error checks applied. Helps define the list of "master or reference tables" used to reconcile parochial definitions and descriptions of terms. Facilitates the ease of adding new source systems.
Foundation for incremental load.	Saves processing time and storage.
Adheres to SMART principle.	Specific, Measurable, Attainable, Relevant, Time bound

Integrated Data Store (IDS)	
Features	**Benefits**
Enforces business rules and constraints.	Data consistency, quality and integrity.
Single source of data about key business functions.	Consistent interpretation of terms and definition. Consistent interpretation of historical data. Provides flexibility in creating several views, data marts, mini marts, etc. for specific data consumers
Foundation for access and security procedures	Assist in defining specific data stewardship procedures. Assist in defining data governance policies and procedures. Provides flexibility in creating several views, data marts, mini marts, etc. for specific data consumers.
Integrated historical data about key business functions and entities.	Different data source systems are integrated having consistent rules and constraints and error checks applied. Facilitates the ease of adding new source systems.
Foundation for incremental load.	Stores historical footprint key business functions and entities. Saves processing time and storage.

Metadata Data Store	
Features	**Benefits**
Business definitions, rules and constraints captured through ORM are able to be used in conjunction with Cognos' ReportNet.	End users of the data have the definitions, rules and constraints that get captured when creating the DSS immediately available. It is easy to lose these during the process, and traditionally these do not get passed through to the end consumers of the data. Allows end consumers the same access to data expertise as was provided to the creators of the system. Passes along tribal knowledge. Increases accurate usage of data and avoids misuse of data in places where data naming conventions do not accurately define the data or the context of the data.

3.2 ORM as a Single Source of Truth for Speed and Agility

ORM as a single source of truth facilitates better communication when creating and supporting decision support systems. This leads to speed and agility as it provides the clarity of "what you're working with" in regard to data. ORM makes it easy to see the business facts and therefore allows teams to sidestep the common pitfall of getting caught in the weeds discussing data nuances in non-productive sessions. Guidant has found this true for initial release of data warehouses and DSS architectures as well as subsequent releases. As business units consume data it is natural to expect significant changes over time as new needs arise and as the level of sophistication in data usage increases. Regardless of whether it is an initial or subsequent release, it takes cross-functional teams to get the work done.

As noted earlier, we have found ORM to be critical for each role across the spectrum of DSS roles and responsibilities. Equally as critical is how ORM supports cross-functional teams of these roles. Consider the following examples where the business facts and ORM conceptual models have driven speed and agility:

- Initial data warehouse population. Data architects, ETL developers, analysts for transactional data, analysts for the analytic data, and application architects meet frequently and on an ongoing basis populate the warehouse. These roles are very detail oriented. With a group of detail-oriented individuals, it is easy to get stuck in non-productive detailed discussions about data without an overarching system such as ORM to provide structure and guidance to the work at hand.
- Successful design and execution. Business solutions designers, data architects, senior leaders, data governance/data stewardship members, application archi-

tects, and training/education experts all need to drive the DSS solution to effectively produce results for the end users at Guidant. Basing design and execution on documented data truths via the ORM conceptual models works to ensure that the DSS solution that gets rolled-out to users meets business needs as planned and that business units can clearly see that needs have been successfully met.

- Post-release evolution of DSS systems. Change is to be expected and is highly desired. Not having requests for change and evolution suggests that the system may not be being used or is not being found valuable. As larger change requests appear, data architects, business solution designers, ETL developers, analysts for transactional and for analytic data, training and education experts, departmental experts, and help desk team members all become involved in discussions. Discussions about the usefulness or the lack of usefulness of data (as perceived by the business users) are well supported by the ORM methodology – ORM supports the clarity needed for productive resolutions.

3.3 Making Use of ORM Meta Data

We have made use of a subset of the ORM meta data to solve some typical challenges that face DSS initiatives. Meta data that is generated through ORM by use of the Visual Studio Enterprise Architect edition (VSEA) tool is parsed into Excel. From Excel the IDS meta data is moved out into the end user DSS tools. From Excel the ODS meta data is moved into the DSS database and then into a reference table application that supports data stewards in their efforts to manage data quality.

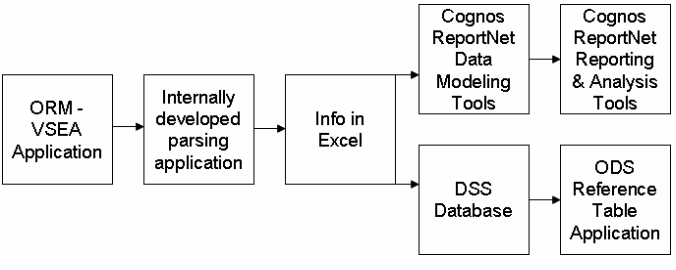


Fig. 5. Illustration of meta data flow

- Meta data into end-user DSS tools. We know that business users frequently do not use the documentation that supports DSS tools (e.g., data dictionaries, models, etc.) – even though it is a best practice for them to do so. They don't always take the time to read documentation when consuming data – which can lead to inaccurate decisions or actions being taken with data. To help reduce this risk, we moved the ORM meta data directly into the end-user tools in addition to providing the external documents. Data definitions are available with the hover of the mouse over data element:

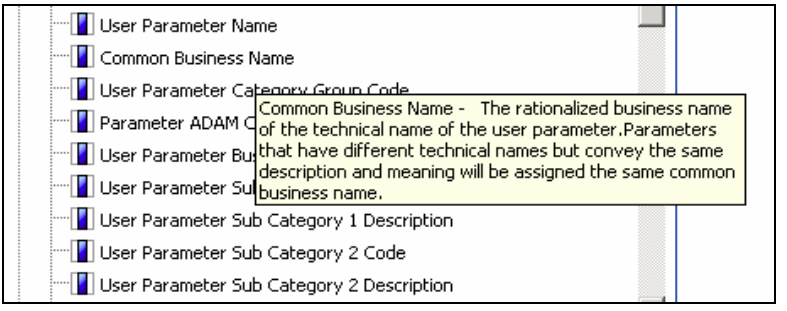


Fig. 6. Illustration of meta data in front end tool

- Data Stewards managing business meta data. Data quality is a constant challenge for any DSS initiative. Having a way to actively manage quality is essential. As part of our data quality solution, we rely on Data Stewards (data experts from the business units that own the source systems). Data Stewards are responsible for maintaining the definitions and descriptions of each data element for which they have assumed ownership. By allowing them to manage the reference tables, the owners of the data can ensure that the data is represented accurately in the DSS system. Consumers of the data are then provided data definitions "right from the source" with less interpretation being done by the DSS implementation teams.



Fig. 7. Meta data in reference table editor tool. Note that the data elements within the reference table editor illustration have been de-identified to protect Guidant's intellectual capital.

4 Conclusion

At Guidant we are keenly aware that our daily business decisions affect lives! Our information systems need to exceed conventional standards, and we are constantly on the lookout for innovative ways drive quality into everything that is Guidant. When developing in-house decision support tools, ORM is one of our corporate best practices.

Requirements Engineering with ORM

Ken Evans

Ambo.biz Ltd, Stickford, Boston, United Kingdom
minden42@onetel.com

Abstract. The number of IT project overspends and failures suggest that many IT projects do not conform to requirements. Despite decades of development the IT industry still seems to lack an effective method of ensuring that a project will be right first time. This paper outlines an ORM based requirements engineering process that aims to reduce the number of IT project failures. The main deliverable of the process is a formal description of WHAT a system is required to do without reference to HOW is to be done. Data or process, which comes first? This paper answers this question by showing how to define processes by starting with an object-role model. To use the approach in this paper you will need the Object-Role Modeling tool embedded within the database function of Microsoft Visual Studio for Enterprise Architects 2003 or later together with two referenced books [Halpin 01] and [Halpin 03].

"Quality is free. It's not a gift, but it is free.
What costs money are the unquality things –
all the actions that involve not doing jobs right the first time."
Philip Crosby [Crosby 79]

1 Introduction

The large number of IT project overspends and failures suggest that many IT projects are of poor quality. In other words – they do not conform to requirements. Despite decades of development projects, the IT industry still seems to lack an effective method of ensuring that a project will be right first time.

For many years there has been a tug-of-war between data and process modellers about which should be defined first, data or process? This paper provides a definitive answer to this question by explaining the principles whereby processes can be defined by using an object-role model as the start point.

The concept of a process as a state machine is not new. However, this paper shows how an object-role model can be used to define the set of permissible state machines (processes) within an arbitrary Universe of Discourse and thus provide the foundation for a provably correct abstract information architecture that provides a formal foundation for an information system.

This paper shows how ORM can contribute to the objective of "getting it right first time." by providing a formal foundation for defining project requirements. This paper outlines a requirements engineering process that can be used to create a formal functional definition which can then be used as a basis for system design. The primary

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 646 – 655, 2005.

deliverable of this requirements engineering process is a formal description of WHAT a system is to do without reference to HOW is to be done.

In this paper "Object-Role Modeling" (ORM) refers to ORM as defined in [Halpin 2001] together with the facilities implemented in the ORM tool embedded within the database function of Microsoft Visual Studio for Enterprise Architects 2003 and explained in [Halpin 03].

1.1 What Is the Business Problem?

Many IT projects either exceed budget and timescale or are cancelled. Tables 1 and 2 are from UK public sector projects reported in The Daily Telegraph 28 June 2005 Page 4. (Multiply costs by 1.8 for US$)

Table 1. Projects completed or scrapped (figures are in millions of pounds sterling)

Project	Estimate	Actual/ Estimate	Overspend	Status
Swanwick air traffic control system	£350	£630	£280	Complete
National Insurance payment system	£170	£260	£90	Complete
Probation office IT system	£85	£120	£120	Scrapped
Individual Learning Accounts	£50	£290	£290	Scrapped
Child Support Agency update	£200	£250	£50	Complete
Magistrates Courts Libra system.	£146	£319	£173	Complete
Passport Agency.	£230	£243	£12	Complete
Totals	£1231	£2112	£1015	

Table 2. Projects in progress or planned

NHS computer system.	£6200	£30000	£23800	WIP
ID cards	£5900	£20000	£14100	WIP
Totals	£12100	£50000	£37900	

The projects in Table 1 show an average overspend of 71.6%. Despite this abysmal track record, estimates for just two of the UK governments "work in progress" IT projects foresee an overspend of 413%. (Table 2. Projects in progress or planned.). These observations suggest that the British Government's lack of a systematic and provably correct way of defining requirements for IT projects is resulting in a huge waste of taxpayers' money.

1.2 What Is the IT Project Problem?

It is clear that project estimates are consistently wrong. Such consistent errors are most probably caused by poor planning. In many cases, an imprecise definition of requirements appears to be followed by poor programme management and the use of questionable development methodologies. Nevertheless, programmers are often put under pressure to produce working code to a politically inspired deadline.

1.3 Requirements Specifications

Some requirements are defined in documents that use a combination of natural language prose and diagrams to define system functions. Most of these documents lack formal precision but nevertheless they must be interpreted by developers who have the responsibility for writing precise code.

The requirements analysis procedures defined in many software engineering textbooks leave much to be desired. For example, Maciaszek recommends the creation of a "System Scope Model" followed by a "Business Use Case Model" and a "Business Class Model" as precursors to UML Use Case Models and Class Models supported by "modern methods of requirements elicitation". [Maciaszek 05]. This advice is misguided because it recommends an informal approach supported by prototyping and informal natural language prose.

Whilst I agree with Maciaszek's statement "*The downstream costs of not capturing, omitting or misinterpreting customer requirements may prove unsustainable later in the process*." [Maciaszek 05 – page 47], the lack of formality in the procedures that he recommends make them not fit for the purpose of defining requirements.

1.4 What Is the Business Need?

"*Quality is conformance to requirements*." [Crosby 79]. The first phase of an IT project should define what the desired system is required to do. Over the last 30 years, several "methodologies" have evolved in an attempt to meet this need. Examples include SSADM, Catalyst, Method/1 and informal and ad-hoc collections of parts of UML. However, if the requirements specifications produced by a methodology do not precisely, unambiguously and formally define the functions of the system to be developed, then they are not fit for the purpose of building software that meets requirements.

2 Hidden Traps in Natural Language

In describing a thing or a situation, you are doing much more than merely registering something that has impinged on one or more of your senses. The act of "describing" *is also an act of classifying*. [Ayer 36].

Hidden within natural language are classification paradigms that have evolved in an ad hoc manner to meet the needs of interpersonal communication. [Kent 98]. Many people assume that the categories in the language they use reflect external reality. However, the Sapir-Whorf hypothesis says that "perceived structures" are really a consequence of the way people use language to classify sensory impressions rather than an accurate reflection of reality. The Sapir-Whorf hypothesis asserts that:

> "The categories and types that we isolate from the world of phenomena we do not find there because they stare every observer in the face; on the contrary, the world is presented in a kaleidoscopic flux of impressions which has to be organized by our minds--and this means largely by the linguistic systems in our minds. We cut nature up, organize it into concepts, and ascribe significances as we do, largely because we are parties to an agreement to organize it this way--an agreement that holds throughout our speech community and is codified in the patterns of our language" [Whorf 56]

The Sapir-Whorf hypothesis has two related principles: 1: *Linguistic determinism:* asserts that the structures and habits of a person's thought are determined by the structure and habits of the person's language. 2: *Linguistic relativity:* asserts that people who speak different languages perceive and think about the world quite differently.

Each generic language (such as English) has sublanguages that reflect the special concepts and habits of thought of the sets of "specialists" who comprise the set of all speakers of the generic language. Thus there is *Engineering English* which is further subdivided into *Mechanical Engineers English*, *Aerodynamic Engineers English*, *Rocket Engineers English* and so on. Boundaries between specialisms are neither defined nor enforced so terms and concepts from one sublanguage tend to be reused (often with slightly or completely different meanings) in other sublanguages. Each sublanguage has its own special words for concepts that only the sublanguage users understand. (e.g. One tribe has 200 words for "snow" and another has 400 words for "cow".)

When formulating requirements, the information analyst may have to interpret sublanguages such as *business English*, *accountants English*, *marketing English* and *public sector English* and even that famous mixture of French and English called *Franglais*. Conceptual and terminological mixing can mislead the reader. For example, French diplomats use the French word "Competencies" to mean "Powers". However, the British Foreign Office English language document that describes the troubled EU Constitution uses the French word "Competencies" rather than the English word "Powers". British people don't mind transferring their "Competencies" but they are up in arms about giving away their sovereign Powers to unelected bureaucrats in a foreign land.

The semantic structures of sublanguages contain implicit concepts that are hidden from those who are unfamiliar with the sublanguage domain. Each word and its related concept may be subliminally linked with predefined attitudinal, emotional and behavioral responses. Thus, hidden semantics affect the way in which speakers of a language think, feel and act.

Most natural language subsets are continually evolving. In the light of the semantic uncertainty that surrounds natural language, it is not surprising that a requirements specification based on natural language produces less than ideal results when used as the basis of a software development project. It is clear that informal natural language is not fit for the purpose of defining requirements for IT projects.

In the light of Whorf's work, we might consider superseding the assertion of René Descartes' "I Think. Therefore I Am." with the post-Whorf assertion: "*I Speak. Therefore I Think*".

2.1 Which Language?

Formal mathematical languages do not suffer from the ambiguities found in informal natural language. For example, since the introduction of algebraic notation (circa 1600), algebra is the same the world over. Thus, it is best to define IT project requirements by using an unambiguous mathematically based language.

2.2 Bridging the Communications Gap

The communications gap between technologists and businessmen is based on more than just problems with jargon. At root, the gap is based on different ways of perceiving the world and different ways of thinking about the world. This gap can be bridged by using a formal language that is able to capture the semantics of an arbitrary Universe of Discourse.

2.3 UML

UML is not a mathematically based language and thus suffers from the same limitations as natural language. As discussed earlier, unconstrained natural language is ambiguous and thus UML is not a suitable tool for defining the requirements of a computer application. UML is presented as an object modeling language for complex systems. However, lack of semantic precision limits UML's effectiveness.

In comparison, the Object-Role Modeling language [Halpin 2001] is a formal, mathematically based language and is thus suitable for specifying information systems requirements. ORM is based on predicate logic which makes fit for the purpose of preparing a formal description of an information system.

3 Cybernetics

Cybernetics is the science of control in man and machine. Cybernetics defines systems in terms of what they do rather than in terms of what they are. The primary question of the cybernetic analyst is "What are all the possible behaviours of this system?"

Cybernetics provides for a scientific treatment of complex systems by using a single set of concepts to represent different kinds of system. Cybernetics can thus be used to compare different kinds of complex systems that at first sight seem to have nothing in common.

Until about 1950, the history of science was characterized by the exploration of simple systems because the traditional scientific approach of "change only one variable at a time" is only possible in simple systems. Cybernetics allows an observer to model simultaneous changes in many variables.

3.1 Cybernetic Transitions

A cybernetic transition defines "what" happens. It does not concern itself with "why" it happens or "how" it happens. It is not necessary to know anything about the mechanisms that perform a set of transitions in order to define an information system. Here are some examples of cybernetic notation from [Ashby 56]:

Transition Assertion: The skin of a white skinned person darkens when exposed to sunshine.

The cybernetic names for the before and after states are "Operand" and "Transform." The thing (process or mechanism) that does the changing is called the "Operator".

Thus: Pale skin (The Operand) is changed by sunlight (The Operator) to dark skin (The Transform).

The set [Operand, Operator and Transform] is called a *transition* and is denoted in the following way:

pale skin ⟶ dark skin.

When we want to describe the behaviour of complex systems we can use sets of simple transitions to define cases where a single operator acts on more than one operand. For example the operator "expose to sunlight" can participate in many transitions, for example:

cold surface ⟶ hot surface

ice ⟶ water

smooth skin ⟶ blistered skin

The principle of closure: When the set of transforms (outputs) contains no element that is not already present in the set of operands (inputs) it is said to be closed. In data terms, this means that all the data types that appear as inputs to a process also appear as outputs from the process. Note that in a closed system, the value of an Operand is not necessarily changed by the Operator so in any given closed transition, the values of one or more operands may be the same as the values in the corresponding transform. The principle of closure is important in cybernetics but an explanation is outside the scope of this paper. See [Ashby 56].

Assumption of discrete change: Cybernetics assumes that all change occurs in discrete steps. This allows all of the important questions to be answered by simple counting.

Every transition can be shown as a matrix: The matrix in Fig 1 represents the transitions:

A → A
B → C
C → B

↓	A	B	C
A	1	0	0
B	0	0	1
C	0	1	0

Fig. 1. A transition matrix

The matrix in Figure 1 can be summarised as shown in Figure 2. Both figures represent the same transition.

	A	B	C
↓	A	C	B

Fig. 2. A summarized transition matrix

Cybernetic notation is used to define what a process does as shown in Fig 3.

This process operates on input A to create output A (A → A) This process operates on input B to create output C (B → C) This process operates on input C to create output B (C → B)

Fig. 3. A simple abstract process description

Fig 3 aims to clarify the meaning of each function within the transition matrix. Fig 4 shows the relationship between a cybernetic transition matrix and a process definition. It also shows how much more convoluted it is to explain a process using natural language prose.

--- PROCESS FUNCTION 1 --- (A → A) Changes input A (defined by the initial value in a specific column in a relational table) to output A (defined by the post-change value in the same column in the same relational table) using the mechanism → --- PROCESS FUNCTION 2--- (B → C) Changes input B (defined by the initial value in a specific column in a relational table) to output C (defined by the post-change value in the same column in the same relational table) using the mechanism → ---PROCESS FUNCTION 3---(C → B) Changes input C (defined by the initial value in a specific column in a relational table) to output B (defined by the post-change value in the same column in the same relational table) using the mechanism →

Fig. 4. Expanded process description

4 Why ORM?

A requirements specification that uses a combination of informal natural language prose and informal diagrams is not fit for purpose. The initial lack of clarity and precision is likely to lead to many changes being requested by users as they are progressively exposed to the inconsistencies in their own thinking processes. The sequence often goes like this: The developer makes a quick start by using the latest RAD fad. (RAD = Rapid Application Development.)

At the end of phase 1 the user looks at the results and says:
"*No! That's not what I said*."
The developer rushes off to update his work.

At the end of phase 2 the user looks at the results and says:
"*No! That's not what I meant.*"
The developer rushes off to update his work.

At the end of phase 3 the user looks at the results and says:
"*No! That's not what I want.*"
The developer rushes off to update his work.

At the end of phase 4 the user looks at the results and says:
"*No! That's not what I need.*"
The developer rushes off to update his work.

At the end of phase 5 the user looks at the results and says:
"*No! That's not what I asked for.*"
The developer rushes off to update his work…

At each stage of this process, the developer's account manager appears with the original contract and says "This is what you contracted for. If you want something else, sign this contract specification change note to authorize the additional expenditure."

Such not-untypical sequences are caused by development activities being started before requirements are properly defined and agreed. Tasks that properly belong in development are misused as part of an evolutionary definition process that everyone hopes will result in a system that conforms to requirements.

Unfortunately, what usually happens is that the RAD fuelled evolutionary learning process is terminated because of budget constraints or to meet politically important deadlines. Developers are then faced with the choice of either canceling the project or shipping what is sometimes termed "good enough" code.

In contrast, a formal isomorphic model is very stable and can be used to help to minimize changes and to reduce project risk. Unfortunately, investments in planning and formal models are often resisted by impatient executives who want to see "results" and see no value in careful planning and abstract models.

4.1 Using ORM to Define Requirements

An Object-Role model can be considered as a container of all possible states of a system. The constraints in an Object-Role model serve to limit the set of possible states to a smaller set of permissible states and to a set of permissible state transitions. The problem of defining "business processes" is simplified to one of defining transitions that are to occur in response to an event.

All IT projects result in a change in the way an organisation or part of an organisation operates. The state before the change is called the AS IS and the state after the change is called the TO BE. Fig 5 depicts an abstract model that is isomorphic to some aspects of both the states of AS IS and TO BE. (The subject of isomorphic models is covered in [Beer 66].)

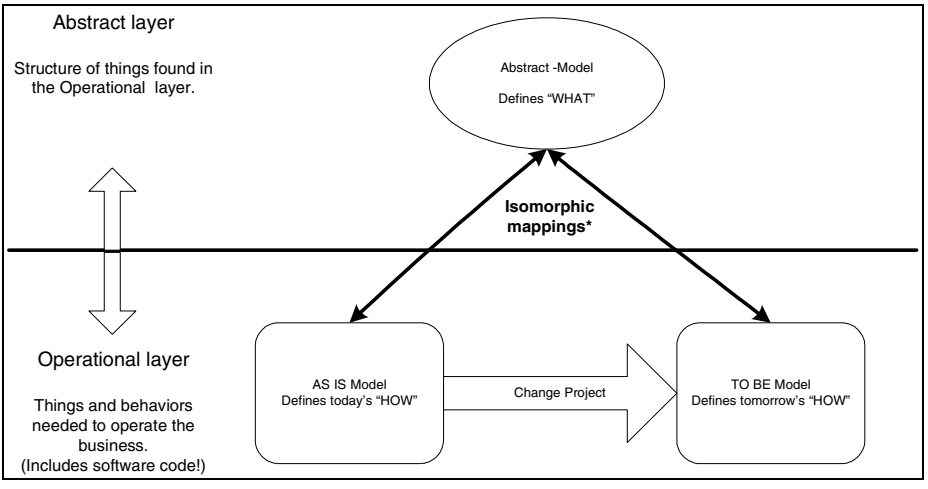


Fig. 5. Abstract models

IT projects that do not use a formalized isomorphic abstract model are more risky than those that do. If a proposed change is not well defined, the tasks required to make the change cannot be well defined. A poorly defined set of tasks makes the job of estimating project duration and project costs little more than guesswork. This is a root cause of variances in project timescales and costs.

An Object-Role model can be used to create a formal abstract model that is isomorphic to key aspects of the organizational states of AS IS and TO BE. In Figure 6, an Object-Role model instantiates the generic concept of the abstract model shown in Figure 5. The abstract layer shown in Figure 6 includes the Object-Role model, the derived 5NF logical model and the transitions (processes).

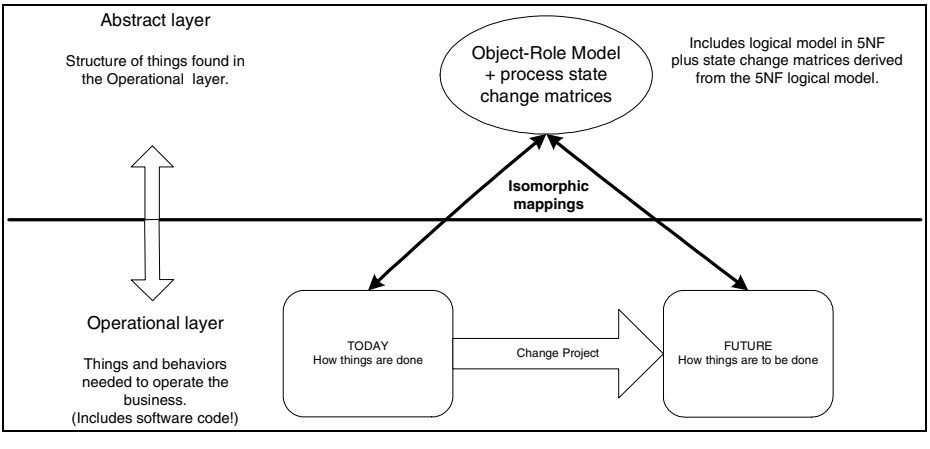


Fig. 6. Using an Object-Role model to define processes

Physical databases are in the "operational layer". Some AS IS databases will not be in 5NF and may contain semantic inconsistencies. However, regardless of the physical structure of the data in any given operational layer, it will always be possible to construct a consistent logical model without duplications.

To use an Object-Role model to define business processes, you first use ORM constraints to define the set of all permissible states and all permissible state transitions. You then generate a 5NF logical model. You then use the columns of the 5NF logical model as the axes of matrices with which you then define event.transition sequences using the cybernetic methods referred to earlier in this paper.

5 Conclusions

Many IT projects are over budget, late or both. The lack of a formal specification of requirements is a root cause of IT project failures.

Many of the methods currently used for requirements analysis and definition are characterized by a lack of formality. Informal methods lead to ambiguous specifications of requirements. Ambiguities in informal natural language together with management pressures create a need for continual rework. These practices are wasteful and unpredictable.

ORM can be used in conjunction with cybernetic principles to create a formal definition of requirements that covers both the data and process aspects of an application. The columns in a fully normalized logical model can be used to define the operands and transforms in a set of transition matrices. ORM tools can thus be used to support the definition and maintenance of a unique set of operands and transforms for an arbitrary Universe of Discourse.

Requirements engineering with ORM can reduce waste and help to deliver projects on time and budget.

References

[Ayer 36] A J Ayer. *Language, Truth and Logic,* Penguin 1936

[Ashby 56] W Ross Ashby, *An Introduction to Cybernetics*, Methuen 1956

[Whorf 56] Benjamin Lee Whorf, *Language, Thought and Reality*, MIT Press 1956

[Beer 66] Stafford Beer *Decision and Control*, Wiley 1966

[Crosby 79] Philip B Crosby, *Quality is Free*, McGraw Hill 1979

[Evans 93] Ken Evans *Reengineering & Cybernetics.* American Programmer Nov 1993.

[Kent 98] *Data and Reality,* 1stBooks 2000 (revised from 1978)

[Halpin 01] Terry Halpin, Information Modeling and Relational Databases, MKP 2001

[Halpin 03] Terry Halpin. Ken Evans, Pat Hallock, Bill Maclean *Database Modeling with Microsoft Visio for Enterprise Architects*. MKP 2003.

[Maciaszek 05] Leszek A. Maciaszek, *Requirements Analysis and System Design,* Pearson, Addison Wesley 2005.

Generating Applications from Object Role Models

Betsy Pepels and Rinus Plasmeijer

Software Technology Department,
Institute for Computing and Information Sciences,
Radboud University Nijmegen, The Netherlands
{betsy, rinus}@cs.ru.nl

Abstract. We propose a generic strategy for generating Information Systems (IS) applications on the basis of an Object Role Model (ORM). This strategy regards an ORM as specifying both static and dynamic aspects of the IS application.

We implemented the strategy in a prototype tool, thereby using state of the art software technology. The tool generates IS applications with a basic functionality.

We regard our strategy as a first investigation of a new way to generate IS applications. Many open and sometimes far reaching research questions arise from this first exploration.

1 Introduction and Motivation

Data models like Object Role Models (ORM's, [1]) are used for a structured development of Information Systems (IS's). After a data model has been set up, it is translated to an implementation scheme, most often a relational schema. This scheme is subsequently the basis of the actual IS implementation.

Many tools exist supporting the development of data models. A substantial part of them automate the translation of the model to the implementation scheme. *Visio for Enterprise Architects* [2] is an example based on Object Role Modeling. To realize subsequently the implementation of the IS application, still a lot of development activities (coding) have to be carried out. This is costly and error prone.

We propose a generic strategy for *generating* IS applications on the basis of an Object Role Model. Automated generation of IS applications is valuable because it reduces development activities substantially, and hence errors and costs. Even more important, when IS development consists of development of models, the IS development process reduces to managing models, offering more control over it.

In our strategy we shift the view on the notion of an ORM. In the classical view, Object Role Modeling is a well-defined *method* for creating an ORM. The resulting ORM defines the data in the UoD and their constraints in a very formal way, thereby specifying a *static* description of its possible populations [3].

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 656–665, 2005.
© Springer-Verlag Berlin Heidelberg 2005

We consider an ORM also to specify *dynamic* aspects of its population. Using both the static and dynamic view on ORM populations, we develop a generic strategy to generate IS applications.

Generating complete IS applications is a pretty over ambitious plan that we certainly can't achieve. To start with, we aim only at giving a *Proof of Concept*, thereby using a very limited class of ORM's. The functionality of the corresponding generated IS applications is accordingly limited.

The core ideas of our strategy are presented in section 2.

As part of the *Proof of Concept*, we constructed a prototype tool. For the actual implementation of our tool, we use the lazy functional programming language Clean [4], developed and implemented by our group. We describe the tool shortly in section 3.

In section 4, we conclude on this first exploration of our new way of IS application generation. Furthermore, we elaborate on the many open and sometimes far reaching research questions that arise from it.

2 Obtaining a Running Application from an ORM

In this section we present the core of our strategy: how to obtain a running IS application from an ORM.

Limitations. In this first approach, we only aim at demonstrating that applications can be generated with our strategy. We use a limited class of ORM's: we do not yet take into account nominalization, subtypes and derived fact types. Furthermore, we limit the possible constraints to only uniqueness constraints (UC's) and mandatory role constraints (MRC's).

We do not cover retrieval functionality yet. In first instance we focus on generating the parts of an IS application that *change* data, in contrast to *retrieving* them. Earlier research has already pointed out that it is very well possible to define query languages directly based on ORM's [5].

Basic Plan. A running application is some definition (a "program") being executed. Hence, to transform some ORM into a running IS application, we have to derive a program definition from that ORM and subsequently execute it.

We consider a running IS application to have basically three ingredients: (1) a fact store (2) functionality (3) a (graphical) user interface (a GUI). We elucidate in subsections 2.1 through 2.3 how we obtain a definition of each of these three ingredients from an ORM. In subsection 2.4, we show how these three definitions are put together and transformed into a running application.

This approach is similar to that of the *Conceptual Information Processor* (CIP) [1]. Our strategy results in a concrete CIP that is inferred from the ORM.

2.1 Deriving the Structure of the Fact Store

In our strategy, we use an IS store that closely matches the structure of an ORM. This differs with the standard approach, in which an ORM is translated to a relational schema.

We associate with the objects and fact types of an ORM data structures capable of holding their population. Because labels do not have a population, we do not associate data structures with them.

With an *object*, we associate a data structure containing the population of the object. With an *n-ary fact type*, we associate a data structure containing a population of n-tuples. The elements of such an n-tuple depend on the roles being part of the fact type. If the role is played by an object, the corresponding tuple element is a *reference* (or *pointer*) to a member of the population of that object. By using references, members of populations can be shared. If the role is played by a label, the tuple element has just the same type as the label. In this way, the fact store resembles some kind of directed graph.

The aim of our different approach is twofold.

First, when generating IS applications from ORM's directly, we dot not want to bother about additional transformations to some operational mechanism, that only serves implementation purposes. So we prefer to have, at least conceptually, a structure of the store not all too different from the ORM itself. If really needed, necessary transformations can be added in a later stage.

The second reason is much more important. By translating to a relational schema, we would limit ourselves unnecessary. Relational schemas allow only some simple types like `Int` and `String`, and sometimes types like `Date`. By using our kind of store, we open the way for richer data types being stored, for instance collection types like `Set` and `Bag`, or special purpose types like `Message`, or recursive types.

2.2 Inferring Functionality

Normally, the functionality of an application is defined separately from the data model. It can for instance be defined by Use Cases. In a running IS application, functionality manifests itself in the user interface.

Conceptually, implementing functionality involves coding all kinds of transformations from the data entered through the user interface to the the actual relational schema, the latter being determined by the data model. These transformations should guarantee the integrity of the data stored.

In our strategy, functionality is not defined separately, but it is inferred (at least partially) from the ORM. This does not only save time and costs, but is also ensures guarantees automatically the integrity of the data stored.

Method. We take a bottom-up approach: we start with deriving properties of populations of ORM's (both static and dynamic) and arrive step-by-step at groups of data that are logically manipulated together by the end user.

Changing the Population of an ORM. The population of an ORM as a whole consists of the single populations of its objects and fact types. At all times, a population must obey the constraints of the ORM. A population may *change*: one or more changes may (simultaneously) be applied to one or more populations of object/fact types. But a change only may be applied if, after the change, the new population doesn't violate the constraints. So constraints

determine the changes allowed to be made to a population. In other words, the constraints of an ORM determine the *operational* or *dynamic* behavior of its population.

Constraints. Normally, in an ORM constraints are used to express properties of the UoD. For this, a collection of standard constraints is available. Also general constraints may be used, often denoted as text. When implementing the corresponding IS application, a few of the standard constraints can be translated to constraints the RDBMS supports, for instance MRC's. The rest somehow has to be implemented by coding. General constraints have to be first interpreted by the developer and then this interpretation has to coded too.

We consider constraints to be *predicates*. Predicates are (*mathematical*) *functions* taking arguments and having as result a `Boolean` value. Here, the arguments are taken from the population of the ORM, and the result of the function must be `True`, otherwise the constraint is violated.

We regard constraints to be *orthogonal* to the objects and fact types of the ORM. Where objects and fact types define the structure of the population of the ORM, constraints limit the actual members of the population.

In our strategy, every constraint in an ORM is to be expressed as a predicate. The developer defines them using the language Clean. This code is used as part of the generated IS application (see below).

Standard constraints are defined once and can be reused subsequently. Every general constraint has to be expressed as a predicate by the developer. This might seem tedious, but there are advantages compared to the traditional way of working. Clean is a language with a very high expressive power, so a constraint can be expressed easier than for instance by coding it in SQL. Furthermore, the developer is forced to state very clearly the exact meaning of the constraint, leaving nothing to the interpretation.

Business Rules. In our vision, business rules are conceptually the same as constraints: they can also be regarded as predicates limiting the actual population of the ORM. Hence they could be added as general constraints to the ORM. There is however a big difference here. General constraints can with some effort be expressed as predicates. Business rules, and certainly the more complex ones, aren't expressed as predicates that easy. In fact, there are two problems here. The first one is how to formalize business rules at all. The second one is to express them as predicates. How to do this in general is subject of future research.

Logical Units of Work. Next, we concretize what a change to an ORM population comprises. A change (or *operation*) to the population of a *single* object/fact type is *implicit*. We recognize three basic operations: element(s) can be added (`add`), or element(s) can be updated (`upd`), or element(s) can be deleted from a population (`del`). Our plan is to infer from an ORM *groups of basic operations associated with objects/fact types* that, when applied simultaneously to populations of those objects/fact types, keep the population of the ORM as a whole valid. Such a group we define to be a *logical unit of work* (an *luw*).

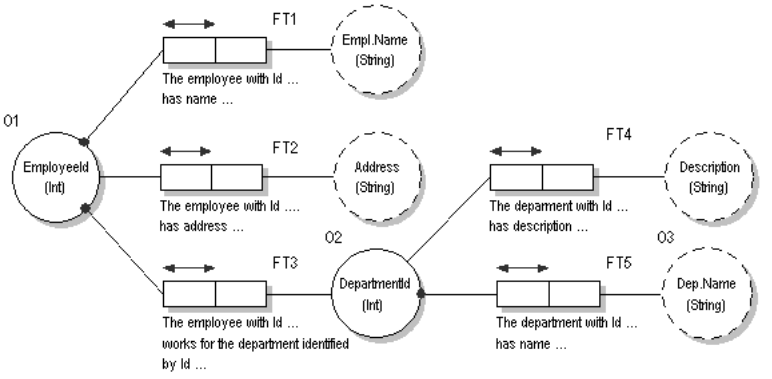


Fig. 1. Example Object Role Model

Examples of Logical Units of Work. In the ORM in figure 1, {`add` on FT1, `add` on FT3, `add` on O1, `add` on O3} but also {`add` on FT2} are *luw*'s.

The term *luw* is chosen because from the ORM point of view, the group of operations is logical in the sense that, when applied simultaneously, it doesn't violate constraints. We will see below that also from the end user point of view an *luw* is logical too. Note that an *luw* applied as a whole keeps the population of the ORM valid; applying only a part of it might turn it invalid. Note further that *luw*'s are defined as operations on the ORM, but have an equivalent for the fact store.

Inferring Logical Units of Work. We use a generic algorithm to obtain *luw*'s from *arbitrary* ORM's. This algorithm is based on *abstract interpretation* [6]. When using abstract interpretation, in stead of reasoning in the concrete domain, it is reasoned in an abstract domain. The aim of abstract interpretation is to derive properties of the concrete domain, by simulating the behavior of the concrete domain in the simpler abstract domain. A detailed description of the abstract interpretation to obtain the *luw*'s is far beyond the scope of this paper and is subject of a separate paper.

The abstract interpretation algorithm takes as input the definition of the ORM and gives as result all possible *luw*'s of the ORM. The algorithm uses rules about constraints and populations, like "*if* a member is `add`'ed to a population of an object, *and* there is a MRC on a role played by that object *and* having a simple UC, *then* also a member *must be* `add`'ed to the fact type containing that role". The abstract interpretation algorithm starts with one object/fact type. It successively gathers all objects/fact types that are involved when changing the population, on basis of the rules. Together they form an *luw*.

Example. Starting with {`add` on FT5}, we find two *luw*'s: {`add` on O2, `add` on FT4, `add` on FT5}, and {`add` on FT5} solely.

Concurrently, the algorithm gathers every constraint that limits the populations of the objects and fact types of this *luw* into a *condition* that must hold if

this *luw* is to be applied. This condition is expressed in the form of a function in Clean, taking as arguments the *actual population* of the ORM and the *actual data to be entered* in the ORM. This function is a *composite* of predicates that were earlier expressed by the developer as a Clean function.

From the abstract interpretation many *luw*'s result, some overlapping. It is up to the developer to point out which *luw*'s will be used for the IS application. *luw*'s can also be joined. Every *luw* used will appear in the running application as a means of manipulating data. Of course, the data the developer wants to be accessible, must be covered somehow in an *luw*.

Real life ORM's with a substantial number of fact types result in an enormous amount of *luw*'s. In this case it is not feasible to let the developer define by hand the set of *luw*'s to be used in the application. Some kind of automated support is needed. This is however unexplored terrain and subject of future research.

Access Model and Functionality. An *luw* involves a group of objects/fact types that is safely manipulated together. An end user has a different view on an *luw*. (S)he is interested in manipulating *data*, not in manipulating *fact types*. This means that the end user only manipulates the "leafs" of the *luw*: only the objects and labels of its constituting fact types.

This group of objects/labels associated with an *luw* we define to be the *access model* of that *luw*. Seen from the end user, an access model is a group of data that are manipulated together and are correlated in some logical way. They may for instance be presented together in a data entry window. This is the way the notion of functionality is correlated with an *luw*.

Example. For the *luw* {`add` on FT1, `add` on FT2, `add` on FT3, `add` on O1, `add` on O2} the access model is formed by O1, O2 and the labels identified by `Address` and `DepartmentId`.

The end user changes data in the form of the access model. These changes cannot be applied directly to the fact store. Therefore we additionally derive transformations (back and forth) between the definitions of the access model and the *luw*'s. These transformations again call the basic operations on the objects/fact types of the *luw*'s.

2.3 Generating the User Interface

To generate the the graphical user interface (GUI), we use the GUI Toolkit [7] that comes as a library with our development environment. The Toolkit works with models. To create an interactive application, it only needs a data model describing *what* what data are to be displayed and a model of *how* it should be displayed, also known as the *view*. The Toolkit is able to translate any change in the view into a change in the data model.

What should be displayed, we derived from an ORM in section 2.2: the access model. This can directly be used. *How* it should be displayed, is up to the developer. (S)he may define the appearance of the access model, for instance: the style, how data is presented (an edit box, drop down boxes) and the layout of the various elements.

2.4 Putting Together and Executing the Definitions

The three definitions we have obtained from an ORM (the structure of the fact store, functionality, the GUI) have to be put together and executed to get a running application. Each of the three definitions is brought to life in its own specific way.

Fact Store. The definition of the structure of the fact store is brought to life in the form of a *dynamic Object Space*. Initially, the Object Space is empty. Subsequently, for each object/fact type in the ORM, a corresponding data structure is created in the Object Space. The populations of these data structures can dynamically be manipulated by primitive access functions (`add`, `del`, `upd`).

We have chosen this approach because of the flexibility it offers: data structures can be added without having to stop and restart the application. In this way, we need only one fact store for storing both the facts defining the application and the facts of the application itself. Furthermore, this makes it possible to change (upgrade) the application while it is running.

Functionality. The definition of functionality is generated in in three parts: the access model, the conditions, and the transformations between the access model and primitive access functions. These parts are generated as functions and stored that way in the fact store.

To properly use these, we built a generic editor that works based on an access model. We create one window for each access model. To get an impression of this editor, see the screen shot of the running application in figure 2. The editor allows entering data in the form of the access model. The definition of the view is used for the actual displaying.

The editor contributes in maintaining the integrity of the data stored. Classically, when changing data in a system, the transaction is rolled back if it turns out that the integrity rules are violated. In our approach, the editor allows data changed in the window *only to be actually stored* if the conditions are satisfied. To check this, the editor calls the function defining the condition.

To put subsequently the data actually in the store, the editor calls the transformations between the access model and primitive access functions, which then are applied to the Object Space.

Graphical User Interface. Our GUI Toolkit creates the interactive part of the IS application, on basis of the access model and the view the developer defined.

3 Prototype Implementation

To test and demonstrate our ideas, we implemented a prototype tool. This prototype is in the first place meant as a research vehicle and not as a real IS application development environment.

For the actual implementation of our tool, we use the functional programming language Clean. The high abstraction level of functional programming languages enables developers to focus on architectural and algorithmical problems in stead

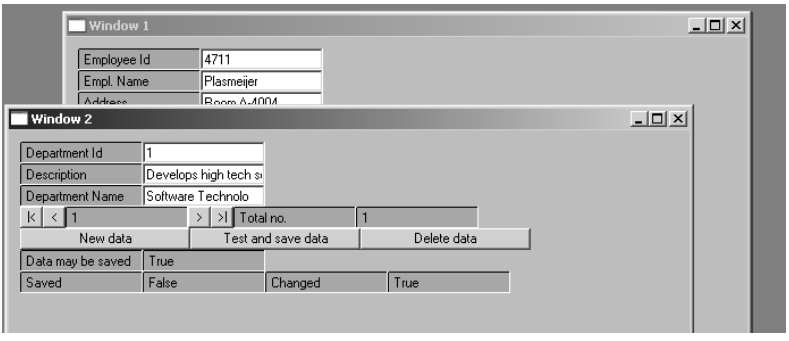


Fig. 2. Screen shot of the running application

of coding. We use its facilities for orthogonal persistence to write the fact store to disk [8].

Architectural Aspects. Every aspect of the IS application generation process is integrated in the tool. In the same environment, both the ORM is defined and the application is executed. The ORM is defined as facts and stored that way in the fact store. Both the intermediate and end results of the translation process and the data of the running application itself are facts as well and stored that way.

Working with the Tool. Using the tool, generating an IS application takes a number of steps. In the first step the developer records the ORM and constraints. In the second step the tool generates from the ORM a *standard* access model and view. In this preliminary version, the developer cannot yet determine for him/herself the composition of *luw*'s and not yet choose a view for them. In the last step, the whole is transformed into a running IS application.

To give an impression, in figure 2 a screen shot of the running application is given. There are two windows each containing the generic editor, one for entering data about employees, and one for data about departments.

4 Conclusion and Future Research

We outlined a generic strategy for IS application generation on the basis of a limited the class of ORM's. This strategy regards an ORM as specifying both static and dynamic aspects of the IS application. We use a fact store for the IS application that is directly correlated to the structure of the ORM. We infer the functionality of the application by generalizing the operational behavior of ORM's using abstract interpretation. This operational behavior is determined by the constraints and business rules of the ORM. The graphical user interface is generated by the GUI Toolkit of our development environment.

In our approach, the developer specifies constraints and business rules statically in the form of predicates. By our way of transforming the ORM into a running application, the dynamic behavior they imply is automatically accomplished.

We built a tool to test and demonstrate our ideas. For the implementation, we used the functional programming language Clean. Even starting from a very limited class of ORM's we are able to generate IS applications with a basic functionality.

Current Limitations and Future Research. The strategy presented here is a very first investigation of this way of IS application generation.

We started from a limited class of ORM's. First of all, we have to work out the strategy for complete ORM's. The abstract interpretation, which we couldn't present here, has to be worked out and described in detail, first for limited ORM's and successively for complete ORM's.

Our presentation of the strategy is an informal one. A more formal and generalized approach is needed, which should be based on, amongst others, operational semantics of ORM's.

The tool should be extended with the possibility of the developer defining the view on the access model.

Many open and sometimes far reaching research questions arise from this first exploration. A first and certainly not exhaustive list includes:

- Constraints and business rules play a central role in our strategy. They are to be expressed as predicates. Research is needed how (complex) business rules can be formalized at all, and how they can be expressed as predicates.
- Our strategy opens the way to have ORM's with labels and objects having richer types than types currently allowed, like collection types and recursive types. It is very promising to research how this is to be defined in an ORM, what the consequences are for IS development using ORM's, how it is to be incorporated in our strategy and what the consequences are for generating IS applications.
- The results of the abstract interpretation involve for practical ORM's large amounts *luw*'s. Research is needed for automated support to handle these.
- Real IS applications have various kinds of dynamic behavior. Behavior in ORM's arises from several sources, like the constraints and business rules and derivable fact types. Most probably, these do not suffice to obtain every desired dynamic behavior of IS applications. This implies that the ORM formalism has to be extended, for instance with explicitly defined flow.
- An IS application evolves. Conceptually, this means that its model changes and that the population has to be adapted to the new model. When IS applications can be completely generated on the basis of a ORM, automated evolution of IS's might become feasible. This is a very promising research theme.

References

1. Halpin, T.: Information modeling and relational databases: from conceptual analysis to logical design. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001)
2. T. Halpin, K. Evans, P. Hallock, B. Maclean: Database Modeling with Microsoft Visio for Enterprise Architects. Morgan Kaufmann (2003)

3. ter Hofstede, A.H.: Information Modelling in Data Intensive Domains. PhD thesis, University of Nijmegen, The Netherlands (1993)
4. Clean home page. (http://www.cs.ru.nl/~clean/)
5. Bloesch, A.C., Halpin, T.A.: Conquer: A conceptual query language. In Thalheim, B., ed.: Conceptual Modeling - ER'96, 15th International Conference on Conceptual Modeling, Cottbus, Germany, October 7-10, 1996, Proceedings. Volume 1157 of Lecture Notes in Computer Science., Springer (1996) 121–133
6. Cousot, P.: Abstract interpretation based formal methods and future challenges, invited paper. In Wilhelm, R., ed.: Informatics — 10 Years Back, 10 Years Ahead. Volume 2000 of Lecture Notes in Computer Science. Springer-Verlag (2001) 138–156
7. Achten, P., van Eekelen, M., Plasmeijer, R., van Weelden, A.: GEC: a toolkit for Generic Rapid Prototyping of Type Safe Interactive Applications. In: 5th International Summer School on Advanced Functional Programming (AFP 2004). To appear in LNCS, Springer (2004) 262–279
8. Vervoort, M., Plasmeijer, R.: Lazy dynamic input/output in the lazy functional language Clean. In Peña, R., Arts, T., eds.: The 14th International Workshop on the Implementation of Functional Languages, IFL'02, Selected Papers. Volume 2670 of LNCS., Springer (2003) 101–117

A Fact-Oriented Approach to Activity Modeling

H.A. (Erik) Proper, S.J.B.A. Hoppenbrouwers, and Th.P. van der Weide

Institute for Computing and Information Sciences, Radboud University Nijmegen,
Toernooiveld 1, 6525 ED Nijmegen, The Netherlands, EU
{E.Proper, S.Hoppenbrouwers, Th.P.vanderWeide}@cs.ru.nl

Abstract. In this paper we investigate the idea of using an ORM model as a starting point to derive an activity model, essentially providing an activity view on the original ORM model. When producing an ORM model of an inherently active domain, the resulting ORM model can provide an appropriate base to start out from. We will illustrate this basic idea by means of a running example. Much work remains to be done, but the results so-far look promissing.

1 Introduction

As argued before [7], ORM is not just suitable for the conceptual design of databases, but for the analysis of domains in general. Even if automation of some functionality is not a goal, ORM can be used to formalize and understand domains.

In this paper we propose to use ORM's fact oriented approach in the context of activity modeling. Addmittedly, the work is still at a preleminary stage. Much further work remains to be done. The basic idea is to provide a smooth transition from an ORM model of an *active* domain to an activity model of that domain. The approach we propose is to indeed start out from a 'plain' ORM model, and then to add 'temporal dependencies' between the fact types in the model. These temporal dependencies are preludes towards the actual triggering relationships between activity types and are therefore based on triggering mechanisms found in workflow modeling approaches. We use YAWL (Yet Another Workflow Language) [1] as a reference point. YAWL is a workflow language which aims to cover a range of workflow patterns that is as wide as possible [1].

Note that the resulting activity model really corresponds to a specific view of the underlying ORM model. Not all details present in an ORM model will appear in an activity model 'view' of the ORM model. Activity models are particularly suited to display the intended *flow* of activities, while an ORM model provides a much more generic perspective on the same domain. Most notably, the rich variety of constraints will not re-appear in the activity model 'view' of an ORM model.

Once the temporal dependencies have been determined, the fact types that are present in the ORM model of an (active) domain can be examined more closely. In this step the aim is to identify the *actors* that perform activities, the *activities* they perform and the *actands* which these activities are performed on.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 666–675, 2005.
© Springer-Verlag Berlin Heidelberg 2005

This leads to an ORM model with roles that are explicitly classified in terms of their *kind* of involvement in activities.

From this attributed ORM model, an activity model can be derived rather mechanically. In follow up research we are looking into ways of further mechanising the derivation of activity models from an attributed ORM model and also to maintain this link during the modeling process. We envisage modelers being able to manipulate the activity models while the underlying ORM domain model is kept up-to-date as well.

Note that the approach suggested does not particularly favor an event-driven, a process-driven or a data-driven modeling approach. ORM is used as a language to model domains in general, starting out from a fact-oriented perspective on the domain. In other words, the domain that is modeled (be it processes, data, or events) is viewed as being represented as a set of *facts*. These facts may deal with processes that occur in a domain or may deal with the kind of data/information that is being manipulated in the domain.

2 Temporal Ordering of Facts

As a running example, consider the following verbalizations (at the type level) of a domain dealing with patients visiting doctors:

A Person fills in a Form
A Doctor produces a Diagnose
A Person is examined by a Doctor
A Doctor writes a Prescription

This leads to the situation as depicted in Figure 1. Suppose now that in this domain we have the case that:

Before a Person can be examined by a Doctor, s/he should have filled in a Form.
Before a Doctor produces a Diagnose, a Person should have been Examined.
Before a Doctor writes a Prescription, a Person should have been Diagnosed.

These rules, however, still provide an incomplete picture. The examination, the diagnosing, and the writing of a prescription should, for a given doctor's visit,

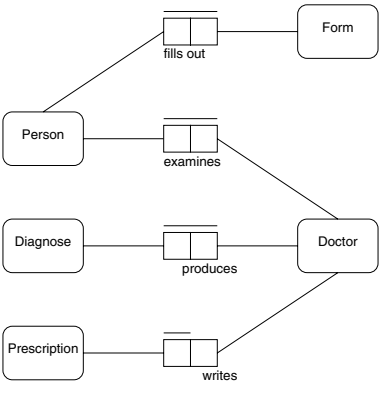


Fig. 1. Basic model of a visit to a Doctor

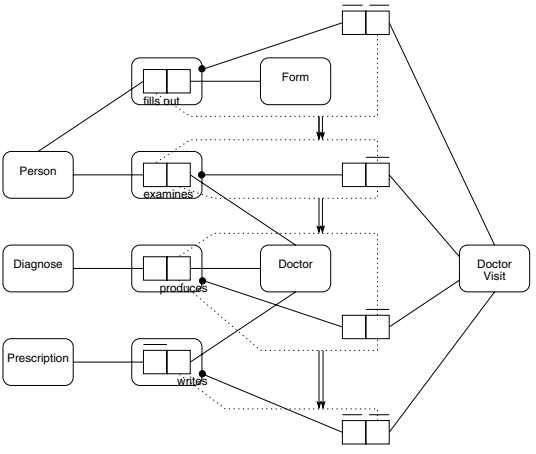


Fig. 2. Doctor visits

all be performed by the same doctor. Even more, as a person may visit a doctor twice for two different reasons, the diagnose and prescription really pertain to a specific doctor visit. This leads to the situation as depicted in Figure 2.

In defining the semantics of the depicted temporal dependencies a time axis is needed. When observing the universe of discourse at point in time t_1 we may observe: Person 'John' fills out Form 'A20.9012', at some later point in time t_2 John may have finished filling out the form. This means that at t_2 we cannot observe Person 'John' fills out Form 'A20.9012'. We could, for the sake of the example, imagine that at t_2 Doctor 'Smith' examines Person 'John' while this was not (yet) the case at t_1.

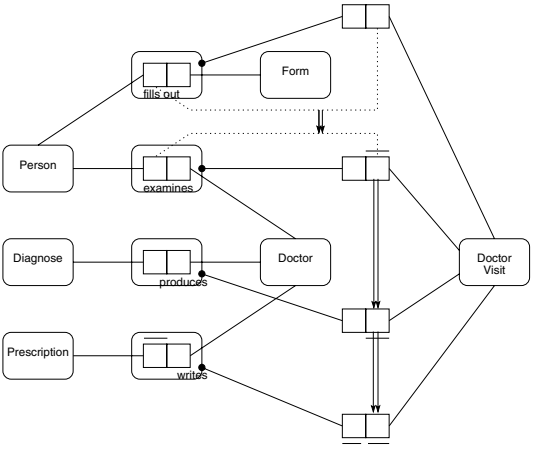


Fig. 3. Doctor visits with alternative semantics

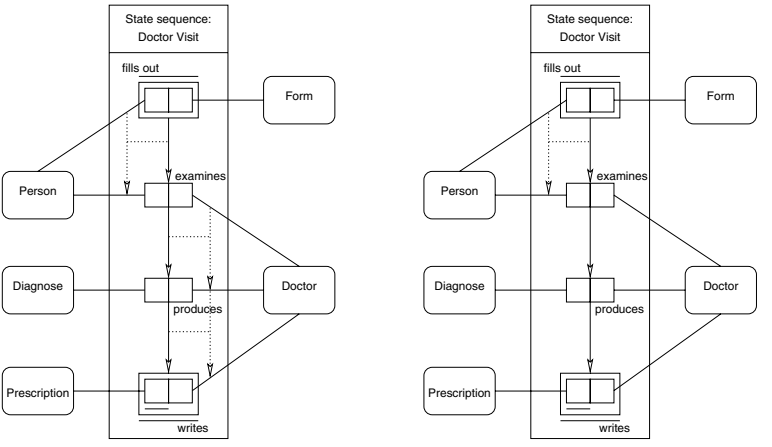


Fig. 4. Compact versions of Doctor visit

Informally, the semantics of the double arrow can now be defined as follows. All role combinations at the source of the arrow should cease to exist (in time) just before role combinations at the destination of the arrow come into existance. A proper formalisation of the semantics of such temporal dependencies is beyond the scope of this paper. It can, however, be defined in terms of the temporal semantics associated to ORM as provided in e.g. [8].

Note that work on *object life cycles* as reported in e.g. [2] focuses on the temporal depencies of roles (in facts) that are played by (the instances of) *one* object type only. However, to be able to arrive at activity models describing the concerted behavior of several objects within a domain, a notation as shown-in / suggested-by Figure 2 is needed.

Now consider the situation as depicted in Figure 3. This time, the doctor who examined the patient does not have to be the doctor producing the diagnose.

Since the diagrams of Figure 2 and 3 are rather complex, we introduce (in line with the notation proposed in [3]) the graphical notation as shown in Figure 4.

Based on the notation used in YAWL [1], the temporal dependencies may be combined to form complex synchronisation patterns. This is illustrated in Figure 5.

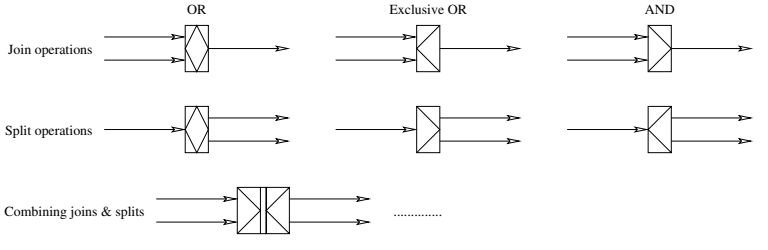


Fig. 5. Complex synchronisation

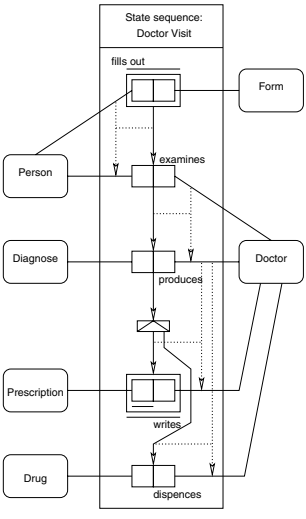


Fig. 6. Choice of medication

An example of the use of a complex synchronisation pattern is provided in Figure 6. In this model, a doctor can either dispense a drug directly, (exclusive) or provide a prescription.

3 Classification of Roles

In activity modeling we are concerned with modeling of domains that are *active*. So there must be *objects* in the domain playing an active *role*. These activities should be reported in the facts that can be verbalized when capturing the universe of discourse. The aim of this section is to take a closer look at these facts in order to find the *activities*, the *actors* who perform them and the *actands* they are performed on. *When* the activities occur is captured by the temporal dependencies as discussed in the previous section.

With these new modeling concepts, we can typically take ORM domain models and "annotate" them in terms of the refined concepts. We will base this process of annotation on linguistic foundations in line with the ORM tradition. Based on these annotated ORM models, we expect to be able to mechanically derive models in a notation that is better suited for the modeling of activities.

Let us now, as an example, consider the following domain:

> A person with name James is writing a letter to his loved one, at the desk in a romantically lit room, on a mid-summer's day, using a pencil, while the cat is watching.

with elementary facts:

> A person is writing a letter
> This person has the name James
> This letter has a romantic nature
> This letter has intended recipient James's loved one
> The writing of this letter by James, occurs on a mid-summer's day

The writing of this letter by James, is done using a pencil
The writing of this letter by James, is done while the cat is watching
The writing of this letter by James, is taking place at a desk
This desk is located in a room
This room is romantically lit

Within these elementary facts, several *players* can be identified. In the above example, we can isolate the players and facts as follows:

[A person] is writing [a letter]
[This person] has [the name James]
[This letter] has a [romantic nature]
[This letter] has intended recipient [James's loved one]
[The writing of this letter by James], occurs on a [mid-summer's day]
[The writing of this letter by James], is done using [a pencil]
[The writing of this letter by James], is done while [the cat] is watching
[The writing of this letter by James], is taking place at [a desk]
[This desk] is located in [a room]
[This room] is lit in [a romantic] way

The writing of the letter is the central fact in the above domain. All players in the facts describing the above domain are players in this domain. What are the activities, actors and actands? Several degrees of activeness exist with regards to the role which a player plays in a fact/domain. Numerous linguistics-oriented frameworks exist to classify the roles that objects play in sentences/facts (for example [5]). In our case we are primarily interested in distinguishing between actors and actands. With this aim in mind, we propose the following classification scheme for roles. The secondary classification level is mainly intended to better and further clarify the top level classification. On a scale of decreasing activity:

Actor role – A role where the player is regarded as carrying out an activity. Linguists may also use the term *agentive*.
In the example domain: The person.
Two sub-classes may be identified:
Initiating role – An agentive role, where the player is regarded as being the initiator of the activity.
Reactive role – A non-initiating agentive role.

Actand role – A role where the player is regarded as experiencing/undergoing an activity.
In the example domain: a letter, a loved one and the cat.
Three sub-classes may be identified:
Patient role – An experiencing role, where the player is regarded as undergoing changes (including its very creation) as intended by the actor.
Beneficiary role – An experiencing role, where the player is regarded as the beneficiary and/or recipient of the results of the activity.

Contextual role – A role where the player is regarded as being a part of the context in which the activity takes place. This role typically corresponds to the "adjuncts" in natural language.
Four sub-classes may be identified:
Instrumental role – A role where the player is regarded as being an instrument in an activity.
In the example domain: a desk and a pencil.
Spatial-locative role – A role, where the player is regarded as being the *physical* location of an activity.
In the example domain: the desk and the room.
Temporal-locative role – A role, where the player is regarded as being a temporal orientation of the activity.
In the example domain: mid-summer's day
Catalysing role – A role, where the presence of the player is regarded as being beneficial (either in a positive or a negative way) to an activity.
In the example domain: the room lit in a romantic way.
Observative role – An experiencing role, where the player is regarded as observing/witnessing the activity
In the example domain: the cat.

Predicative role – A role where the player is regarded as being a predicate on some other player.
In the example domain: the name James.

Each role in an elementary fact must fit within one of these classes. The choice between the different classes is subjective. It depends on the viewer. An elementary fact is an *activity* if-and-only-if it contains at least one actand or actor role, otherwise it is a *predication*. The objects playing the four main classes of roles are regarded as the *actors*, *actands*, *context elements*, and *predicators* respectively.

For the example given above, we would have:

Activity: [Actor: A person] is writing [Actand: a letter]
Predication: [This person] has [the name James]
Predication: [This letter] has a [romantic nature]
Predication: [This letter] has intended recipient [James's loved one]
Predication: [The writing of this letter by James], occurs on a [mid-summer's day]
Predication: [The writing of this letter by James], is done using [a pencil]
Predication: [The writing of this letter by James], is done while [the cat] is watching
Predication: [The writing of this letter by James], is taking place at [a desk]
Predication: [This desk] is located in [a room]
Predication: [This room] is lit in [a romantic] way

In the example domain, the writing of the letter by the person is regarded as the key activity in the domain. In other words, writing is an activity, while the person is the actand and the letter is the actand. We may regard the cat and the loved one as an actand as well. What about the pen, the name James, the desk, etc? They are really players in *predications* over the other players. The fact that a pen is used by the person to write the letter is a *predication* of the writing *activity*.

If we were to zoom in on a sub-domain of the above sketched domain, we could actually find that what is an actand in the super-domain is an actor in the sub-domain. Consider, for example, the sub-domain:

[The writing of this letter by James], is done while [the cat] is watching

When considered in isolation, one may quite easily argue that the primary activity here is the watching, which is something that is being done by the cat. This really makes the cat into an actor rather than an actand, while the thing that is being watched (the writing) becomes the actand. This means that our notions of actor, actand, activity and predication are really to be taken *relative* to the domain under consideration.

As mentioned before, the work reported is still in its preleminary stages. One of the work that remains to be done is the provision of a sound theoretical base by which modelers can determine wether an object is a actor, actand or activity. In doing so, we plan on integrating the work on DEMO [9], which has a sound theoretical base in speech-act theory [4].

When taking our example domain of doctor visits we have:

Activity: [Actor: Person] fills out [Actand: Form]
Activity: [Actor: Doctor] examines [Actor: Person]
Activity: [Actor: Doctor] produces [Actand: Diagnose]
Activity: [Actor: Doctor] writes [Actand: Prescription]

Note that we have chosen to regard the person, who is being examined, as an actor as well. It is an example of a *collaborative* activity. Graphically, we now obtain the situation as depicted in Figure 7. Note that the classification is associated to the *roles*. Even though the example does not show it, an object

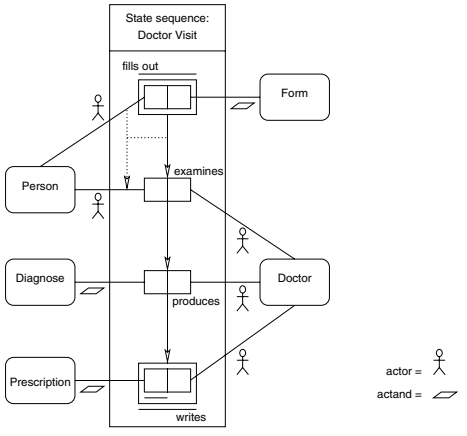


Fig. 7. Attributed ORM model of a Doctor visit

may (for obvious reasons) be an actor in one activity, while it is an actand in another one.

Finally, Figure 8 depicts the two variations of the doctor visit example domain as an activity model using the ArchiMate [6] notation. The notation used in this diagram is (arguably) better suited to represent activity models than the

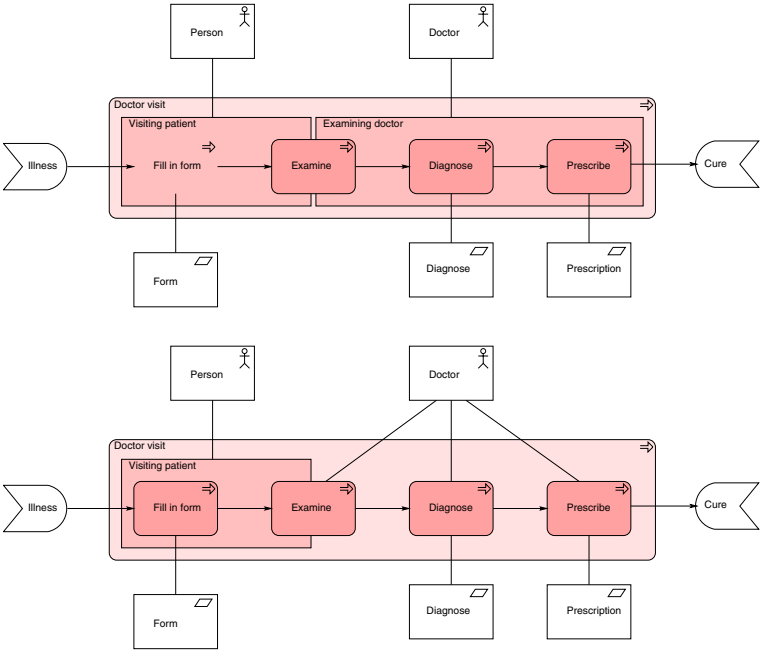


Fig. 8. Doctor visit as an activity model

ORM notation. However, we have now established a clear (and formalizable) relationship between activity models as presented in Figure 8 and ORM models as shown in Figure 4. Note the one-on-one correspondence between the actors, actands and activities.

4 Conclusions

In this short paper we have presented the idea of using ORM's fact oriented approach in the context of activity modeling. The basic idea as presented, is to provide a smooth transition from an ORM model of an *active* domain to an activity model of that domain. We did so by adding temporal dependency constraints to the ORM model, and attributing the model with a classification of roles. The resulting model was mapped onto a graphical notation used for activity modeling in a mechanical manner.

As mentioned before, the resulting activity model really corresponds to a specific view of the underlying ORM model. Not all details present in an ORM model will appear in an activity model 'view' of the ORM model. Activity models are particularly suited to display the intended *flow* of activities, while an ORM model provides a much more generic perspective on the same domain.

In future research, we will be looking at ways to provide modelers with more guidelines on classifying the roles. Furthermore, the relationship between the workflow patterns from YAWL [1] and the temporal dependencies that may be used in ORM models needs further investigation. We also intend to look at strategies to further mechanise the derivation of activity models from an attributed ORM model and also to maintain this link during the modeling process. We envisage modelers being able to manipulate the activity models while the underlying ORM domain model is maintained as well.

Finally, using ORM as a generalized domain modeling approach, and then 'specializing' this model towards an activity model (as illustrated in this paper), is not an idea that is limited to activity modeling alone. The general underlying idea of using a fact-oriented approach to produce an ORM model for a given domain, and then to specialize this model by re-interpreting the fact types in terms of a specialized classification, is an approach that is expected to be suitable for a wide range of specialized modeling languages including business modeling and architecture modeling. We will therefore also investigate these workings in more detail.

References

1. W.M.P. van der Aalst and A.H.M. ter Hofstede. YAWL: yet another workflow language. *Information Systems*, 30(4):245–275, 2005.
2. P. van Bommel, P.J.M. Frederiks, and Th.P. van der Weide. Object-Oriented Modeling based on Logbooks. *The Computer Journal*, 39(9):793–799, 1996.
3. P.N. Creasy and H.A. (Erik) Proper. A Generic Model for 3-Dimensional Conceptual Modelling. *Data & Knowledge Engineering*, 20(2):119–162, 1996.

4. J. Habermas. *The Theory for Communicative Action: Reason and Rationalization of Society*, volume 1. Boston Beacon Press, Boston, Massachusetts, 1984.
5. J.J.A.C. Hoppenbrouwers, B. van der Vos, and S.J.B.A. Hoppenbrouwers. Nl structures and conceptual modelling: Grammalizing for KISS. *Data & Knowledge Engineering*, 23(1):79–92, 1997.
6. M.M. Lankhorst and others. *Enterprise Architecture at Work : Modelling, Communication and Analysis*. Springer, Berlin, Germany, EU, 2005.
7. H.A. (Erik) Proper, A.I. Bleeker, and S.J.B.A. Hoppenbrouwers. Object-role modelling as a domain modelling approach. In J. Grundspenkis and M. Kirikova, editors, *Proceedings of the Workshop on Evaluating Modeling Methods for Systems Analysis and Design (EMMSAD'04), held in conjunctiun with the 16th Conference on Advanced Information Systems 2004 (CAiSE 2004)*, volume 3, pages 317–328, Riga, Latvia, EU, June 2004. Faculty of Computer Science and Information Technology, Riga Technical University, Riga, Latvia, EU.
8. H.A. (Erik) Proper and Th.P. van der Weide. EVORM - A Conceptual Modelling Technique for Evolving Application Domains. *Data & Knowledge Engineering*, 12:313–359, 1994.
9. V.E. van Reijswoud, J.B.F Mulder, and J.L.G. Dietz. Commucation Action Based Business Process and Information Modelling with DEMO. *The Information Systems Journal*, 9(2):117–138, 1999.

ORM 2

Terry Halpin

Neumont University, Salt Lake City, Utah, USA
terry.halpin@neumont.edu

Abstract. Object-role Modeling (ORM) is a fact-oriented modeling approach for specifying, transforming, and querying information at a conceptual level. Unlike Entity-Relationship modeling and Unified Modeling Language class diagrams, fact-oriented modeling is attribute-free, treating all elementary facts as relationships. For information modeling, fact-oriented graphical notations are typically far more expressive than other notations. Introduced 30 years ago, ORM has evolved into closely related dialects, and is supported by industrial and academic tools. Industrial experience has identified ways to improve current ORM languages (graphical and textual) and associated tools. A project is now under way to provide tool support for a second generation ORM (called ORM 2), that has significant advances over current ORM technology. This paper provides an overview of, and motivation for, the enhancements introduced by ORM 2, and discusses an open-source ORM 2 tool under development.

1 Introduction

Object-Role Modeling (ORM) is a fact-oriented approach for modeling, transforming, and querying business domain information in terms of the underlying facts of interest, where all facts and rules may be verbalized in language readily understandable by non-technical users of those business domains. Unlike Entity-Relationship (ER) modeling [6] and Unified Modeling Language (UML) class diagrams [31, 32, 33], ORM treats all facts as relationships (unary, binary, ternary etc.). How facts are grouped into structures (e.g. attribute-based entity types, classes, relation schemes, XML schemas) is considered an implementation issue irrelevant to capturing business semantics. Avoiding attributes in the base model enhances semantic stability and populatability, and facilitates natural verbalization. For information modeling, fact-oriented graphical notations are typically far more expressive than other graphical notations. Fact-oriented textual languages are based on formal subsets of native languages, so are easier to understand by business people than technical languages like the Object Constraint Language (OCL) [35]. Fact-oriented modeling includes procedures for mapping to attribute-based structures, such as those of ER or UML. For a basic introduction to ORM, see [14], and for a thorough treatment see [15].

Though less well known than ER and object-oriented approaches, fact-oriented modeling has been used successfully in industry for over 30 years, and is taught in universities around the world. The fact-oriented approach comprises a family of closely related "dialects", some using the generic term "Object-Role Modeling" (ORM) [15], and some using different names such as Natural language Information Analysis Method (NIAM) [12, 36], and Fully-Communication Oriented Information

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 676 – 687, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Modeling (FCO-IM) [1, 2]. ORM languages include RIDL [30], LISA-D [26, 27] and FORML [15]. Though using a different notation, the Object-oriented Systems Model (OSM) [11] is a close relative to ORM, with its attribute-free approach.

Commercial tools supporting the fact-oriented approach include the ORM solution within Microsoft's Visio for Enterprise Architects [23], and the FCO-IM tool CaseTalk [5]. Free ORM tools include VisioModeler [34] and Infagon [28], as well as various academic prototypes. Dogma Modeler [10], an ORM-based tool for specifying ontologies, is currently being significantly extended.

Industrial ORM experience has identified ways to improve current ORM languages (graphical and textual) and tools. Our project aims to specify and provide tool support for a second generation ORM (called *ORM 2*), that has significant advances over current ORM technology in both functionality and usability. This paper overviews and motivates several enhancements introduced by ORM 2, and discusses our tool under development to support it. The initial development team comprised of faculty and students at Neumont University is being expanded to include external collaborators from industry and academia. The current implementation is coded in C# as a free, open-source plug-in to Microsoft Visual Studio .NET, using the new Microsoft Designer Framework Software Development Kit for building domain specific languages.

The rest of this paper is structured as follows. Section 2 focuses on improvements made to the ORM graphical notation. Section 3 discusses enhancements to the ORM textual notation. Section 4 discusses tooling aspects. Section 5 summarizes the main results, suggests topics for further research, and lists references. An online appendix includes sample schemas in the new notation, and a screen shot from the new tool.

2 The ORM 2 Graphical Notation

This section includes sample diagrams to contrast the new notation with the current notation used by the ORM source model solution in Microsoft Visio for Enterprise Architects [23]. The main objectives for the ORM 2 graphical notation are:

- More compact display of ORM models without compromising clarity
- Improved internationalization (e.g. avoid English language symbols)
- Notation changes acceptable to a short-list of key ORM users
- Simplified drawing rules to facilitate creation of a graphical editor
- Full support of textual annotations (e.g. footnoting of textual rules)
- Extended use of views for selectively displaying/suppressing detail
- Support for new features (e.g. role path delineation, closure aspects, modalities).

Although far more expressive graphically than UML or industrial ER for static data models, ORM schema diagrams typically consume more space because of their attribute-free nature (which also leads to greater semantic stability). The larger diagram size problem may be ameliorated by providing attribute-views on demand, and/or by redesigning the ORM graphic notation to be more compact. The first solution includes displaying "minor fact types" as attributes on an ORM diagram, and automatically generating attribute-based schemas for implementation (e.g. relational schemas, OO class schemas, XML schemas). As we have not yet completed our implementation of attribute view toggles, we instead focus on the new ORM graphical notation we have

implemented. This notation is more compact, typically yielding diagrams about 65% the size of an equivalent diagram in the old notation. All English-specific symbols in the old notation have been replaced by language-neutral symbols to improve localization in different language communities. A survey was issued to eighteen ORM experts. Each change introduced by the new notation was acceptable to a majority.

Most of the ORM 2 figures in this paper were drawn using an ORM2 Visio stencil that our team created. Currently, the stencil may be used for drawing purposes only. In contrast, the ORM 2 modeling tool is intended to support automatic transformation between graphical and textual representations, as well as transformation to/from other schemas (e.g. relational, class, and XML schemas), and code generation (e.g. to DDL or program code). The tool is also intended to front-end other modeling tools (e.g. one might enter and transform an ORM schema to a relational schema for export to another database design tool to generate the DDL code). As schemas developed in the tool are fully exposed as XML, there is significant scope for inter-operability.

2.1 Object Type Shapes

In the old ORM notation, object type (entity type and value type) shapes are depicted by named ellipses. Ellipses are faster for users to draw manually, but they consume more space than rectangles (hard or soft), especially when names are long. For ORM 2, the default shape for object types is a *soft rectangle* (rectangle with rounded corners). Besides providing a more compact container for the enclosed text, this is consistent with the current notation for nested object types. The shape auto-sizes to provide appropriate white space around the text. Users may spread text over multiple lines (as in the Visio ORM source solution). Text is displayed in a user-definable default style, individual text elements may be user-selected for alternate styles, and text may be left/center/right justified.

To make this notation change more acceptable, we allow an ellipse or a hard rectangle as an alternative shape for object types, as set by a configuration option. Fig. 1 shows some examples. If the ellipse option is chosen, the shape is still more compact than the old notation. Object type shape examples in the rest of this document use the default shape (soft rectangle). Of the 18 experts in the survey, 12 preferred the soft rectangle, 5 preferred the ellipse shape, and one preferred the hard rectangle.

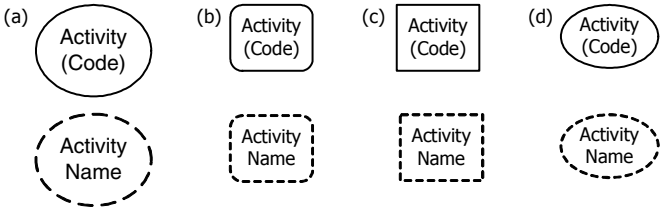


Fig. 1. Object type shapes in (a) old and new notations: (b) default: (c) and (d) alternatives

2.2 Shapes and Readings for Predicates and Roles

To save space, the *size of the role boxes is significantly reduced* (see Fig. 2). A line connecting a role box to an object type shape goes from the mid-point of an outer

edge of the role box to the *center* of the object type shape (unlike the old solution, which uses connection points). *Predicate readings* may be user-positioned beside the predicate shape. Although it is no longer possible to place a predicate reading inside a role box, as in Fig. 2(a), the role boxes may now be used to include role sequence details (to disambiguate role paths). By default, all text is in 7 point Tahoma (Visio uses 8 point Arial). Of the 18 experts surveyed, 14 approved the changes.

The reduced role box size allows multiple single-role set-comparison constraints, (Fig. 2(d)). Unlike the old notation in Fig. 2(c), to add an extra set-comparison constraint between the role-pairs the new notation requires moving a constraint to make room for the third constraint, but since such cases are rare this minor inconvenience is acceptable. The size of the constraint bubbles is slightly reduced in the new notation.

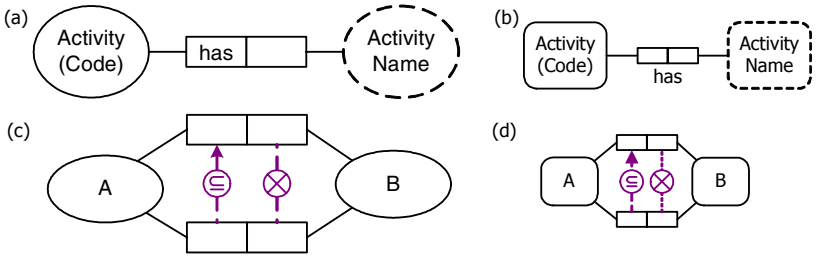


Fig. 2. The role box and text size in ORM (left side) is reduced in ORM 2 (right side)

A forward predicate reading is read left-to-right or top-to-bottom, and an inverse predicate reading (pre-pended by "« ", or possibly an arrow-head "◀") is read right-to-left or bottom-to-top. Two binary predicate readings may be displayed together, separated by a slash, or separately on either side of the predicate shape with the inverse reading pre-pended by "« ". The display of any predicate reading may be toggled on/off. Multi-line reading displays are allowed. Fig. 3(b) shows some possibilities.

For a fact type with n roles ($n > 0$), ORM 2 allows predicate readings for all possible ($n!$) permutations of role orderings. For each such role ordering, one or more *alias readings* may be supplied (e.g. "is employed by" as an alias of "works for"). Query navigation in relational style from any role of an n-ary fact type is enabled by just n predicate readings (one starting at each role), but industrial modelers requested this additional flexibility. For non-binary fact types, at most one predicate reading is displayed on the diagram. ORM 2 allows a name (as distinct from a reading) for the fact type (e.g. "Birth" for Person was born on Date), though this is not normally displayed on the diagram. One use of *fact type names* is to generate a suitable target name for fact types that map to a single table or class. Multi-line fact type names are allowed.

The *display of role names* in square brackets (Fig. 3(c)) may be user-positioned and toggled on/off. Multi-line role names are allowed. The display toggle may be set globally or on an individual role basis. Although each fact type has at least one predicate reading, the display of predicate readings may be suppressed (e.g. to focus on role names). By default, role names are displayed in a different color (e.g. indigo).

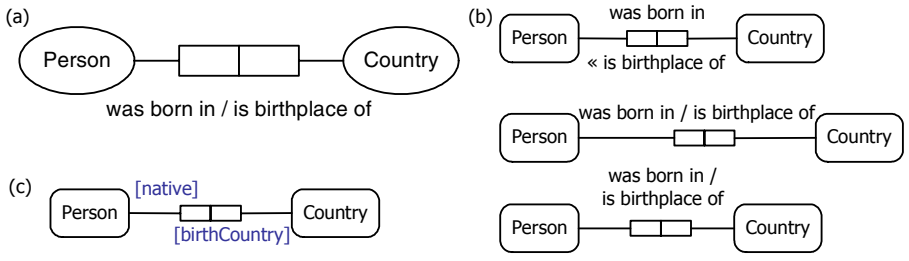


Fig. 3. Predicate and role readings display in (a) ORM and (b), (c) ORM 2

Apart from size reduction, *objectification* in ORM 2 allows *nesting of unary predicates*, as well as *predicates with non-spanning uniqueness constraints*, in accordance with new formal semantics and guidelines for objectification for ORM 2 [16, 22].

2.3 Uniqueness Constraints

Internal uniqueness constraints apply to single predicates, and appear in ORM as arrow-tipped lines. ORM 2 instead uses *simple lines*, which are faster to draw manually, and intuitively correspond to the common practice of underlining keys (Fig. 4(b)). The line is shorter than the role box length to avoid ambiguity. The old ORM notation marks a primary uniqueness constraint as "P". In ORM 2, a *preferred uniqueness constraint* is indicated by a *double line* (as in the practice of doubly underlining primary keys when alternate keys exist). This also avoids the English bias of "P". In ORM 2 the notion of preferred uniqueness is conceptual (a business decision to prefer a particular identification scheme). By default, all ORM 2 constraints are colored violet.

In the case of an internal uniqueness constraint that spans non-contiguous roles, a *dashed line* bridges the gap across the inner role(s) that are excluded from the constraint. Such a case may arise only if the association is ternary or higher. For example, the upper uniqueness constraint on the ternary in Fig. 4 spans the first and last roles. Of the 18 experts surveyed, 17 preferred the new internal constraint notation.

An *external uniqueness constraint* spans roles from different predicates. The old ORM notation depicts this by a circled "U" (for unique), or "P" (for "primary") if used for primary reference (Fig. 4(a)). This notation is biased towards English, and differs totally from the internal uniqueness notation. For localization and consistency, the new notation (Fig. 4(b)) uses a *circled underline*, or a *circled double underline* if the constraint provides the preferred identification scheme (consistent with the new internal uniqueness constraint notation and the horizontal notation for relational schemas [15]). Of the 18 experts surveyed, 14 preferred this new constraint notation.

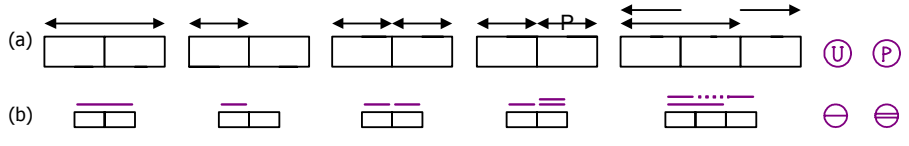


Fig. 4. Uniqueness constraints in (a) ORM and (b) ORM 2

2.4 Mandatory Role Constraints

In the old ORM notation, *simple mandatory role constraints* are indicated by a solid dot either (a) at the intersection of an entity type shape and the line connecting it to a role, or (b) at the role end. Option (b) avoids ambiguity when an object type plays many mandatory roles whose connections to the object type are too close to distinguish the role to which the dot applies. *Disjunctive mandatory (inclusive-or) constraints* are depicted by placing the solid dot in a circle connected by dotted lines to the roles it applies to. ORM 2 retains this notation, except that the solid dot is consistently colored *violet* and a global *configuration option* determines the default placement of simple mandatory dots at the role or object type end. Users may override this global setting on an individual role basis. Fig. 5 shows some simple examples.

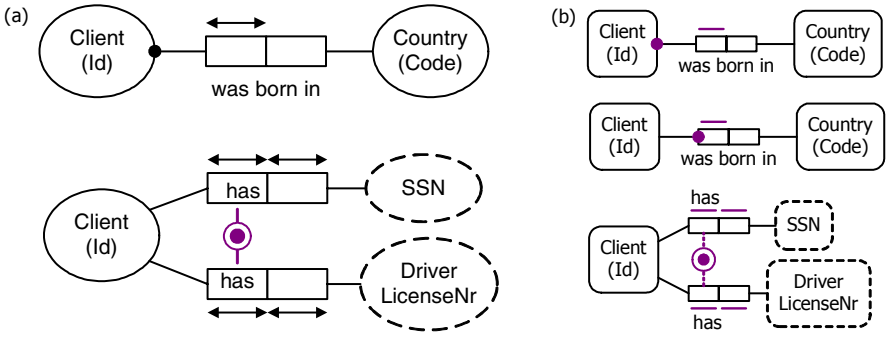


Fig. 5. Mandatory constraints in (a) ORM and (b) ORM 2

2.5 Set-comparison, Exclusive-or, Frequency and Value Constraints

Subset, exclusion, and equality constraints continue to be denoted by a circled ⊆, ×, = respectively, connected to the relevant roles by dashed lines, except that the ORM 2 shapes are a bit smaller, with refined symbols. In addition, ORM 2 supports the *n-ary version of the equality constraint*. As usual, *exclusive-or* constraints are denoted by combining the circled × with the circled dot, and users may display the two component constraints overlaid or separately (as in Visio). Fig. 6(b) shows the basic shapes.

Frequency constraints are displayed as in Visio, except that single symbols (≤, ≥) replace double symbols (<=, >=), for example 2, ≥3, 2..5. *Value constraints* are denoted as in Visio, except that many values may be displayed on a single line (e.g. {'M', 'F'}, {'a',..'z'}), and *open ranges* are supported (e.g. "> 0" for PositiveReal).

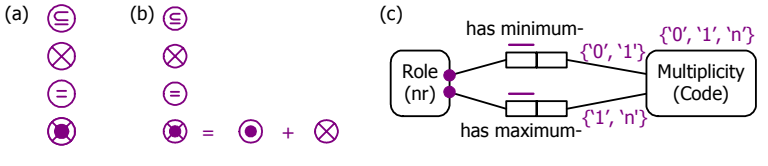


Fig. 6. Set-comparison, Xor and value constraints

ORM 2 allows *value constraints on roles*, not just object types. For example, the Information Engineering metamodel fragment in Fig. 6(c) includes value constraints on the minMultiplicity and maxMultiplicity roles. Of the 18 experts surveyed, all favored support for open ranges, and 17 favored role-based value constraints.

2.6 Ring Constraints and Subtyping

The current Visio ORM tool uses English abbreviations for various *ring constraints*: ir = irreflexive, as = symmetric, ans = antisymmetric, it = intransitive, ac = acyclic, sym =symmetric. Ring constraints are displayed as a list of one or more of these options, appended to a ring symbol "o", and connected to the two relevant roles (if placed very close, the connection display is suppressed). For example, the reporting relationship is declared to be acyclic and intransitive as shown in Fig. 7(a).

This old notation has disadvantages: the abbreviations are English-specific (e.g. Japanese readers might not find "ir" to be an intuitive choice for irreflexivity); and the ring constraint display tends to cross over other lines (as in the example). To remove the English bias, and suggest the constraint semantics, ORM 2 uses *intuitive icons*. To reduce edge crossings, ORM 2 *omits role links* if the predicate has just two roles played by the same object type (or compatible object types). For example, in ORM 2 the reporting relationship is declared acyclic and intransitive as shown in Fig. 7(b).

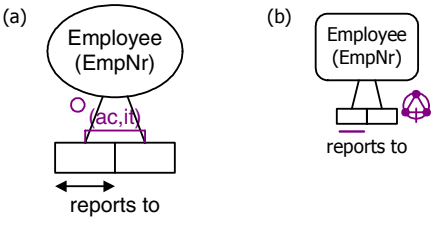


Fig. 7. Acyclic and Intransitive ring constraints depicted in (a) ORM and (b) ORM 2

The ORM 2 icons for ring constraints are loosely based on our icons for teaching ring constraints [15, sec. 7.3], where small circles depict objects, arrows depict relationships, and a bar indicates forbidden (Fig. 8). The different node fills in the antisymmetric icon indicate that the inverse relationship is forbidden only if the objects differ (in the other icons, the objects may be the same). For diagramming purposes, these teaching icons take up too make room, especially when combinations of ring constraints apply.

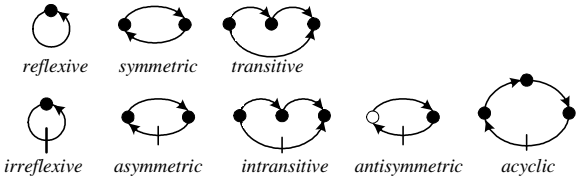


Fig. 8. The original icons used for teaching ring constraints

So simplifying adaptations were made to ensure the final icons are distinguishable while maintaining a compact footprint. The ORM 2 icons (Fig. 9) print clearly at 600 dpi, and are readable on screens at typical resolutions used for industrial modeling. They may be distinguished on low resolution screens by increasing the zoom level. ORM 2 has an icon for each of the ten simple or combined ring constraint choices supported by the current ORM source model solution. In contrast to the teaching icons, arrow-heads are removed (they are assumed), and relevant pairs of constraints are collapsed to a single icon. While the simplified icons are less intuitive than the teaching icons, once their origin is explained it should be easy to remember their meaning. Of the 18 experts surveyed, 14 agreed to the new ring constraint icons.

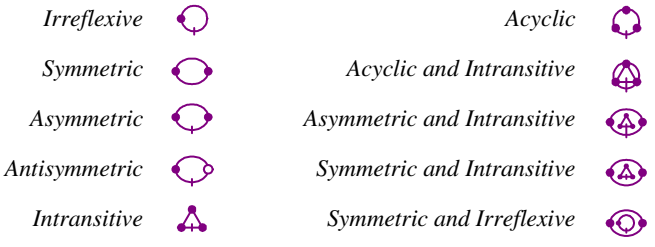


Fig. 9. ORM 2 icons for ring constraints

The current arrow notation for *subtyping* will remain, perhaps supplemented by Euler diagrams as an alternative display option for simple cases. ORM 2 adds support for explicit display of *subtype exclusion* (⊗) *and exhaustion* (⊙) *constraints*, overlaying them when combined, as shown in Fig. 10. As such constraints are typically implied by other constraints in conjunction with subtype definitions, their display may be toggled on/off. Of the 18 experts surveyed, 17 approved this extension.

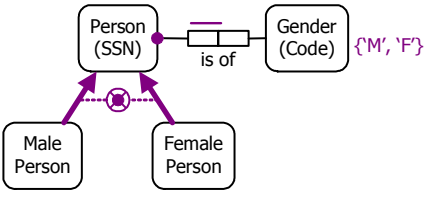


Fig. 10. Explicit display of exclusion and exhaustion constraints for a subtyping scheme

3 The ORM 2 Textual Notation

ORM 2 will support a high level, formal, *textual language* for the declaration of ORM schemas (including fact types, constraints and derivation rules), ORM queries, and possibly fact addition and deletion. The tool will generate code to implement the semantics conveyed by the textual language. The textual language will be cover all the semantics conveyed by the graphical language, as well as additional semantics (e.g. constraints that cannot be captured graphically). All graphical constraints will have

automated *constraint verbalizations*, using improvements discussed elsewhere [19]. Unlike the Visio ORM solution, the ORM 2 textual language may be used to input models, instead of being merely an output language for verbalizing diagrams.

Textual constraints may be noted on the diagram by *footnote numbers*, with the textual reading of the constraints provided in *footnotes* that can be both printed and accessed interactively by clicking the footnote number. Fig. 11 provides an example. Of the 18 experts surveyed, 17 approved the use of footnotes for textual constraints.

Derivation rules may be specified for derived object types (subtype definitions) and derived fact types. ORM 2 allows *fully-derived subtypes* (full subtype definition provided), *partly-derived subtypes* (partial subtype definition provided) and *asserted subtypes* (no subtype definition provided). *Subtype definition*s will be supported as derivation rules in a high level formal language rather than mere comments, and may be displayed in text boxes as footnotes on the diagram. Iff-rules are used for full derivation, and if-rules for partial derivation. Here are sample rules in both ORM 2's textual language and predicate logic for fully and partly derived subtypes respectively:

Each Australian **is a** Person **who** was born in Country 'AU'.
$\forall x$ [Australian $x \equiv \exists y$:Country $\exists z$:CountryCode (x was born in y & y has z & z = 'AU')]

Person$_1$ **is a** Grandparent **if** Person$_1$ is a parent of **some** Person$_2$ who is a parent of **some** Person$_3$.
$\forall x$:Person [Grandparent x $\subset \exists y$:Person $\exists z$:Person (x is a parent of y & y is a parent of z)]

The final grammar for the textual language is not yet determined, but should support declaration of ORM models and queries in *relational style*, *attribute style* and *mixed style*. Relational style uses predicate readings (e.g. the subtype definitions above), while attribute style uses role names. Attribute style is especially useful for derivation rules and textual constraints of a mathematical nature (e.g. see Fig. 11).

As an example of a derivation rule for a derived fact type, we may define the uncle association in relational style thus: Person$_1$ is an uncle of Person$_2$ **iff** Person$_1$ is a brother of **a** Person$_3$ **who** is a parent of Person$_2$. Assuming the role names "brother" and "parent" are declared, we may also specify the rule in attribute style thus: **For each** Person: uncle = brother **of** parent. Further examples may be found elsewhere [19, 21].

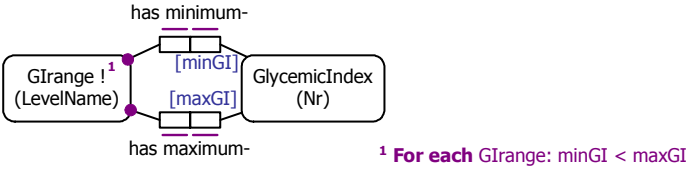


Fig. 11. Textual constraints appear as footnotes in ORM 2

4 Tooling Aspects

This section briefly considers further aspects of the ORM 2 tool. A screen shot from the tool illustrating use of Intellisense in the fact editor is accessible online at www.orm.net/orm2/dietschema.doc. There is no space here to detail the tool's features, but the tool should significantly extend the functionality of Microsoft's

current ORM tool, both by supporting ORM 2 and by adding features. For example, reference modes are treated as views of reference fact types, and the tool allows automatic toggling of the display between the implicit (reference mode) and explicit (reference fact type) representations. This toggle capability is being extended to support attribute-based displays as views of the underlying fact-based schemas. Full integration with Visual Studio provides a powerful and familiar environment for developers, and the architecture and open source nature facilitates adaptation and extensions by others.

The ability to hide/show types of constraints by placing them on different layers should be at least as versatile at that in the Visio solution, which offers 5 layers. The ORM 2 tool will probably extend this to allow suppressing display of all constraints (including internal uniqueness and mandatory constraints).

One abstraction mechanism provided by the Visio ORM solution is the ability to spread a model over many pages, where each page focuses on a sub-area of the business domain. Model elements may be redisplayed multiple times on different pages, as well as on the same page (to reduce edge-crossings). The Show-Relationships feature is indispensable for revealing connections (and for other reasons). At a minimum, this functionality should be supported. Ideally, users should be able to decompose models into multiple levels of refinement, and abstract upwards, with the tool automating the process via the major object type algorithm [3, 15] or a similar technique.

5 Conclusion

This paper discussed a project under way to specify and provide open source tool support for a second generation ORM (called ORM 2), that provides significant advances over current ORM technology. Proposed changes to the graphical notation were described, and their motivation explained. Results from a survey of ORM experts with regard to these changes were noted. Various enhancements to the ORM textual notation were examined, and some improvements from a tooling perspective were identified. To better appreciate the difference made by the new notation, a three page Diet model in both the Visio ORM source model notation and the ORM 2 notation is available as an online appendix at http://www.orm.net/orm2/dietschema.doc.

Parties who own a copy of Visio (Standard edition or higher) and who wish to explore the new graphical notation using models of their own may download a zip file containing the Visio ORM 2 stencil and template plus a sample model file from the following site: www.orm.net/ORM2_Beta.zip. This ORM 2 stencil is for drawing only—it does not generate code. Parties who wish to collaborate in the actual development of the open source ORM 2 tool should e-mail the author of this paper.

The tools project team is currently researching extensions to ORM in several areas, including richer support for join constraints (e.g. distinguishing inner-outer join semantics, displaying complex join constraints, and role path disambiguation [20]), extended open/closed world semantics, and deontic/alethic constraint distinctions [19].

Acknowledgements

This paper benefited from discussion with the tools project team at Neumont University, and from the ORM 2 survey responses by Don Baisley, Dick Barden, Scott Becker, Linda Bird, Anthony Bloesch, Necito dela Cruz, Dave Cuyler, Lance Delano, Jan Dietz, Ken Evans, Gordon Everest, Brian Farish, Pat Hallock, Bill MacLean, John Miller, Borje Olsson, Erik Proper and Pieter Verheyden.

References

1. Bakema, G., Zwart, J. & van der Lek, H. 1994, 'Fully Communication Oriented NIAM', *NIAM-ISDM 1994 Conf. Working Papers*, eds G. M. Nijssen & J. Sharp, Albuquerque, NM, pp. L1-35.
2. Bakema, G., Zwart, J. & van der Lek, H. 2000, *Fully Communication Oriented Information Modellin*g, Ten Hagen Stam, The Netherlands.
3. Bird, L., Goodchild, A. & Halpin, T.A. 2000, 'Object Role Modeling and XML Schema', *Conceptual Modeling – ER2000*, Proc. 19th Int. ER Conference, Salt Lake City, Springer LNCS 1920, pp. 309-322.
4. Bloesch, A. & Halpin, T. 1997, 'Conceptual queries using ConQuer-II', *Proc. ER'97: 16th Int. Conf. on conceptual modeling*, Springer LNCS, no. 1331, pp. 113-26.
5. CaseTalk website: http://www.casetalk.com/php/.
6. Chen, P. P. 1976, 'The entity-relationship model—towards a unified view of data'. *ACM Transactions on Database Systems*, 1(1), pp. 9–36.
7. Cuyler, D. & Halpin, T. 2005, 'Two Meta-Models for Object-Role Modeling', *Information Modeling Methods and Methodologies, eds* J. Krogstie, T. Halpin, & K. Siau, Idea Publishing Group, Hershey PA, USA (pp. 17-42).
8. Demey J., Jarrar M. & Meersman R. 2002, 'A Markup Language for ORM Business Rules', in Schroeder M. & Wagner G. (eds.), *Proc. International Workshop on Rule Markup Languages for Business Rules on the Semantic Web* (RuleML-ISWC02 workshop), pp. 107-128, online at http://www.starlab.vub.ac.be/publications/STARpublications.htm.
9. De Troyer, O. & Meersman, R. 1995, 'A Logic Framework for a Semantics of Object Oriented Data Modeling', *OOER'95, Proc. 14th International ER Conference*, Gold Coast, Australia, Springer LNCS 1021, pp. 238-249.
10. DogmaModeler: www.starlab.vub.ac.be/research/dogma/dogmamodeler/dm.htm.
11. Embley, D.W. 1998, *Object Database Management*, Addison-Wesley, Reading MA, USA.
12. Falkenberg, E. D. 1976, 'Concepts for modelling information', in Nijssen G. M. (ed) *Proc 1976 IFIP Working Conf on Modelling in Data Base Management Systems*, Freudenstadt, Germany, North-Holland Publishing, pp. 95-109.
13. Halpin, T. 1989, 'A Logical Analysis of Information Systems: static aspects of the data-oriented perspective', doctoral dissertation, University of Queensland.
14. Halpin, T. 1998, 'ORM/NIAM Object-Role Modeling', *Handbook on Information Systems Architectures*, eds P. Bernus, K. Mertins & G. Schmidt, Springer-Verlag, Berlin, pp. 81-101.
15. Halpin, T. 2001, *Information Modeling and Relational Databases*, Morgan Kaufmann, San Francisco.
16. Halpin, T. 2003, 'Uniqueness Constraints on Objectified Associations', *Journal of Conceptual Modeling*, Oct. 2003. URL: www.orm.net/pdf/JCM2003Oct.pdf.

17. Halpin, T. 2004, 'Comparing Metamodels for ER, ORM and UML Data Models', *Advanced Topics in Database Research, vol. 3*, ed. K. Siau, Idea Publishing Group, Hershey PA, USA, Ch. II (pp. 23-44).
18. Halpin, T. 2004, 'Information Modeling and Higher-Order Types', *Proc. CAiSE'04 Workshops*, vol. 1, (eds Grundspenkis, J. & Kirkova, M.), Riga Tech. University, pp. 233-48. Online at http://www.orm.net/pdf/EMMSAD2004.pdf.
19. Halpin, T. 2004, 'Business Rule Verbalization', *Information Systems Technology and its Applications*, Proc. ISTA-2004, (eds Doroshenko, A., Halpin, T. Liddle, S. & Mayr, H), Salt Lake City, Lec. Notes in Informatics, vol. P-48, pp. 39-52.
20. Halpin, T. 2005, 'Constraints on Conceptual Join Paths', *Information Modeling Methods and Methodologies, eds J.* Krogstie, T. Halpin, T.A. & K. Siau, Idea Publishing Group, Hershey PA, USA, pp. 258-77.
21. Halpin, T. 2005, 'Verbalizing Business Rules: Part 11', *Business Rules Journal*, Vol. 6, No. 6. URL: http://www.BRCommunity.com/a2005/b238.html.
22. Halpin, T. 2005, 'Objectification', *Proc. CAiSE'05 Workshops, vol. 1*, eds J. Calestro & E. Teniente, FEUP Porto (June), pp. 519-31.
23. Halpin, T., Evans, K, Hallock, P. & MacLean, W. 2003, *Database Modeling with Microsoft® Visio for Enterprise Architects*, Morgan Kaufmann, San Francisco.
24. Halpin, T. & Proper, H. 1995, 'Database schema transformation and optimization', *Proc. OOER'95: Object-Oriented and Entity-Relationship Modeling*, Springer LNCS, vol. 1021, pp. 191-203.
25. Halpin, T. & Wagner, G. 2003, 'Modeling Reactive Behavior in ORM'. *Conceptual Modeling – ER2003*, Proc. 22nd ER Conference, Chicago, October 2003, Springer LNCS.
26. ter Hofstede, A. H. M., Proper, H. A. & Weide, th. P. van der 1993, 'Formal definition of a conceptual language for the description and manipulation of information models', *Information Systems*, vol. 18, no. 7, pp. 489-523.
27. ter Hofstede A. H. M, Weide th. P. van der 1993, 'Expressiveness in conceptual data modeling', *Data and Knowledge Engineering*, 10(1), pp. 65-100.
28. Infagon website: http://www.mattic.com/Infagon.html.
29. Lyons, J. 1995, *Linguistic Semantics: An Introduction*, Cambridge University Press: Cambridge, UK.
30. Meersman, R. M. 1982, *The RIDL conceptual language*. Research report, Int. Centre for Information Analysis Services, Control Data Belgium, Brussels.
31. Object Management Group 2003, *UML 2.0 Infrastructure Specification*. Online: www.omg.org/uml.
32. Object Management Group 2003, *UML 2.0 Superstructure Specification*. Online: www.omg.org/uml.
33. Rumbaugh J., Jacobson, I. & Booch, G. 1999, *The Unified Language Reference Manual*, Addison-Wesley, Reading, MA.
34. VisioModeler download site: http://www.microsoft.com/downloads/results.aspx?displaylang=en& freeText=VisioModeler.
35. Warmer, J. & Kleppe, A. 1999, *The Object Constraint Language: precise modeling with UML*, Addison-Wesley.
36. Wintraecken J. 1990, *The NIAM Information Analysis Method: Theory and Practice*, Kluwer, Deventer, The Netherlands.

A World Ontology Specification Language

Jan L.G. Dietz

Delft University of Technology,
Chair of Information Systems Design
j.l.g.dietz@ewi.tudelft.nl

Abstract. A language is proposed for the specification of the ontology of a world. Contrary to current ontology languages, it includes the transition space of a world, in addition to its state space. For the sake of a clear and deep understanding of the difference between state space and transition space, two kinds of facts are distinguished: stata (things that are just the case) and facta (things that are brought about). The application of the language is demonstrated using a library as the example case.

1 Introduction

After the introduction of the Semantic Web [1], a growing interest in ontologies and in ontology languages has originated. They serve to specify the shared knowledge among a community of people and/or artificial agents [8]. Several languages have been proposed, of which OWL [14] is probably the best known. The contribution of this paper to the field of ontology is mainly a theoretical one, although a very concrete ontology language is proposed too. As has been stated and discussed in [4, 2, 6], the original notion of ontology covers all aspects of a system. With every system a world can be associated in which the effects of the actions of the system take place. Although this notion of world is quite similar to the notion of Universe of Discourse (UoD) [7] we prefer the term "world" in order to avoid any confusion with the conceptual schema of (the database of) a UoD. Moreover, we will not only model the state space of a world, as the current ontology languages only do, but also its transition space. This constitutes the justification for proposing the ontology language as presented in this paper.

In section 2, those notions are discussed that we consider to be most fundamental for understanding what the factual knowledge about a world actually is. This study will lead to a precise definition of the notion of world ontology, based on a distinction between two types of facts: stata and facta. In conformity with this definition, the grammar of a generic language for specifying world ontologies is presented in section 3. It is called WOSL (World Ontology Specification Language). Its semantic basis as well as its diagrammatical notation is adopted from ORM [9]. The application of WOSL is demonstrated in section 4, taking the common operations in a public library as the example case. Section 5 contains the conclusions and some suggestions for further research regarding the relationship between WOSL and ORM.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 688 – 699, 2005.
© Springer-Verlag Berlin Heidelberg 2005

2 Factual Knowledge

2.1 The Ontological Parallelogram

By factual knowledge we mean knowledge about the state and the state changes of a world, like knowing that a person or a car or an insurance policy exists, as well as knowing that the insurance policy of a car started at some date. The basis for understanding factual knowledge is the meaning triangle [10], as exhibited in Fig. 1. It shows the three core notions in semiotics and their relationships. It explains how people use signs as representations of objects in order to be able to communicate about these objects in their absence, i.e., when they cannot be shown or pointed at.

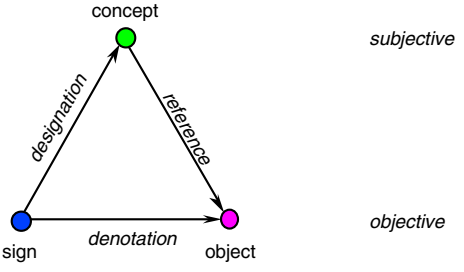


Fig. 1. The meaning triangle

The elementary notions that we will make use of, are designated by the words "sign", "object" and "concept". The notion of concept is considered to be a subjective notion whereas sign and object are considered to be objective notions. *Objective* means that it concerns things outside[1] the human mind[2]. Whether an object is a sign is not an inherent property of the object but something attributed; it is the outcome of an agreement between the subjects that use the sign for their communication. *Subjective* means that it concerns things that can only exist inside the human mind. The three notions are elaborated below.

A *sign* is an object that is used as a representation of something else. A well-known class of signs are the symbolic signs, as used in all natural languages. Examples of symbolic signs are the person name "Ludwig Wittgenstein", the car license number "16-EX-AF", and the numeral "3", as well as the sentences "Ludwig Wittgenstein owns the car with license number 16-EX-AF" and "the car with license number 16-EX-AF is 3 years old".

An *object* is an objective thing, an observable and identifiable individual thing, for example a person or a car. An object is by definition something in the objective realm. To exemplify the notion of objective: objects are things that exist or could exist even

[1] This does not imply that we adopt the objectivist philosophical stance. In fact, we adopt the constructivist stance. So, the objective reality is viewed as a constructed inter-subjective reality [11]. We will not elaborate our philosophical stance.

[2] By "outside the human mind" we actually mean outside the current thought. So, it is possible to think of (other) thoughts; these thoughts then constitute the objective reality.

if there were no human minds (except for the case that thoughts figure as objects). We say that the perceivable properties of an object collectively constitute the 'form' of the object. Objects may be composite: an aggregation of two or more objects is also an object. A car is a good example of a composite object.

The notion of object as explained above is actually the notion of *concrete object*. Only concrete objects are observable by human beings. However, there are many interesting objects that are not observable. The number 3 for example or the composite object denoted by "Ludwig Wittgenstein owns car 16-EX-AF". These objects are called *abstract objects*. The notion of abstract object is nothing more or less than a convenient way of analogous reasoning such that there is an object connected to every concept (see below).

A *concept* is a subjective individual thing. It is a thought or mental picture of an object that a subject may have in his or her mind. A concept is by definition typed: it is always and inevitably a concept of a type. This is just a consequence of how the human mind works. Classification is a very basic conceptual principle, reflected in all natural languages by the linguistic notion of noun: nouns represent types. We will come back to this shortly. In line with the distinction between concrete and abstract objects, we will also speak of concrete and abstract concepts. Examples of concrete concepts are the mental picture I have of the person Ludwig Wittgenstein, and the mental picture you may have of the car with license number 16-EX-AF. Examples of abstract concepts are the number 3, the fact that Ludwig Wittgenstein owns car 1 6-EX-AF, and the fact that this car is 3 years old.

The basic notions of sign, object and concept are related to each other by three basic notional relationships (cf. Fig. 1): designation, denotation, and reference.

Designation is a relationship between a sign and a concept. We say that *a sign designates a concept*. Examples: the name "Ludwig Wittgenstein" designates a particular concept of the type person; the numeric code "16-EX-AF" designates a particular concept of the type car license; the numeral "3" designates a particular concept of the type number.

Denotation is a relationship between a sign and an object. We say that *a sign denotes an object*. Examples: the name "Ludwig Wittgenstein" denotes a particular object, viz. the object of the person Ludwig Wittgenstein; the numeric code "16-EX-AF" on the plate of a car denotes a particular object, viz. the object of that particular car. Analogously, we say that the numeral “3” denotes the abstract object 3 etc.

Reference is a relationship between a concept and an object: *a concept refers to an object*. Examples: the concept Ludwig Wittgenstein refers to a particular person; the concept of the car with license number 16-EX-AF refers to a particular car. Similarly, the concept 3 refers to a particular (abstract) object.

A *type* is a subjective thing. Examples: the type person, the type car, the type number, the type owns, the type age. The human mind applies types in observing the outside world. One cannot do otherwise. Types may be viewed as to operate as 'prescriptions of form' [13]. This 'prescription of form' is also called the *intension* of the type. The 'form' of an object may conform to one or more types, giving rise to one or more (individual) concepts. For example, a material object has a shape, is of a particular material, and has a color. Consequently, one such object can be referred to by three individual concepts, each of a different type, e.g. a cube, a wooden thing and a green thing. Also, the ‘form’ of the object, denoted by “Ludwig Wittgenstein” may

conform to the type person, but also to the type teacher or philosopher or patient, and it may do so simultaneously.

A *class* is a collection of objects. By definition a class contains all objects that conform to the associated type. Examples: the class of persons, i.e., the collection of objects that share those properties that make them conform to the type person, the class of cars, i.e., the collection of objects that conform to the type car, and the collection of object pairs <person, car> that share the property that the person owns the car.

Extension is a relationship between a type and a class. We say that *a class is the extension of a type*. Examples: the class persons is the extension of the type person; the class cars is the extension of the type car; the class ownerships is the extension of the type owns.

The relationships between individual concepts and generic concepts (types), and consequently between individual objects and classes are depicted in Fig. 2. In this figure, which is based on Fig. 1, we have deliberately left out the signs (predicate names and proper names) because they are not relevant in ontology. Ontology is about the essence of things, not about how we name them. The resulting figure is called the *ontological parallelogram*. It explains how (individual) concepts are created in the human mind (cf. [12, 13]). The notional relationships instantiation conformity, and population are explained hereafter.

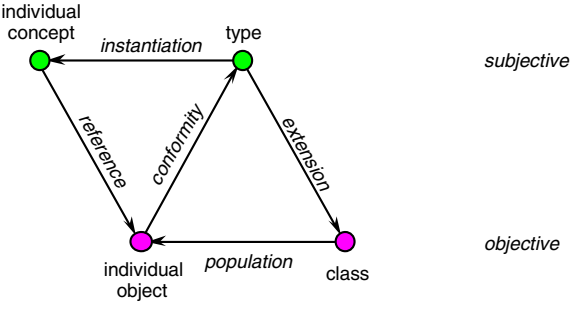


Fig. 2. The ontological parallelogram

Instantiation is a relationship between a concept and a type: *every concept is an instantiation of a type*. Examples: the person Ludwig Wittgenstein is an instantiation of the type person; the car 16-EX-AF is an instantiation of the type car; the number 3 is an instantiation of the type number; the concept Ludwig Wittgenstein owns car 16-EX-AF is an instantiation of the type owns.

Conformity is a relationship between (the 'form' of) an object and a type. We say that *an object conforms to a type*. Examples: the object, denoted by the sign "Ludwig Wittgenstein", conforms to the type person; the object, denoted by the sign "16-EX-AF", conforms to the type car. Both objects are concrete objects. As said already, we like to apply analogous reasoning with regard to abstract objects. So, we also say that the object that is denoted by the numeral "3", conforms to the type number.

Population is a relationship between an object and a class. We say that *a class is a population of objects*. A more common way of expressing this is saying that

the object is a member of or belongs to the class. Examples: the object, denoted by the sign "Ludwig Wittgenstein", belongs to the class persons; the object, denoted by the sign "16-EX-AF", belongs to the class cars; the object denoted by "Ludwig Wittgenstein owns car 16-EX-AF", belongs to the class of ownerships.

2.2 Stata and Facta

The ontological parallelogram, as presented above, will be the basis for developing the ontology of a world. At any moment a world is in a particular *state*, which is simply defined as a set of objects; these objects are said to be current during the time that the state prevails. A state change is called a *transition*. A transition is simply defined as an ordered pair of states. E.g., T1 = <S1,S2> is the transition from the state S1 to the state S2. The occurrence of a transition is called an *event*. An event therefore can simply be defined as a pair <T,*t*>, where T is a transition and *t* is a point in time. Consequently, a transition can take place several times during the lifetime of a world, events however are unique. An event is caused by an act (Note. We will not elaborate in this paper on the relationship between act and event. For a more extensive discussion of acts and events, the reader is referred to [6]).

In order to understand profoundly what a state of a world is, and what a state transition, it is necessary to distinguish between two kinds of objects, which we will call stata (singular: statum) and facta (singular: factum). A *statum*[3] is something that is just the case and that will always be the case; it is constant. Otherwise said, it is an inherent property of a thing or an inherent relationship between things. Examples of stata in the context of a library are the ones expressed in the next assertive sentences (Note: the variables, represented by capital letters, are place holders for object instances):

"the author of book title T is A"
"the membership of loan L is M"

The existence of these objects is timeless. For example, a particular book title has a particular author (or several authors). If it is the case at some point in time, it will forever be the case. One might even say that it has always been the case, only not knowable before some point in time (namely before the book was written). The concept of statum seems to be a useful concept, as opposed to factum (see below). We will therefore adopt this distinction in the language WOSL. A similar remark holds for derived stata. A derived statum is defined by its derivation rule. The being specified of this rule is the only necessary and sufficient condition for the existence of the derived statum. This marks an important difference between a world and a database system about that world. E.g. the age of a person in some world is just there (does just exists) at any moment; in the corresponding database system however, it has to be computed when it is needed. Stata are subject to *existence laws*. These laws require or prohibit the coexistence of stata. For example, if the author of some book is "Ludwig Wittgenstein", it cannot also be "John Irving".

[3] The word "statum" is a Latin word, derived from the intransitive verb "stare" of which the meaning is "to stand", " to be".

Contrary to a statum, a *factum*[4] is the result or the effect of an act. Examples of facta in the context of a library are the ones expressed in the next perfective sentences (Note: the variables are again place holders for object instances):

"book title T has been published"
"loan L has been started"

The becoming existent of a factum is a transition. Before the occurrence of the transition, it did not exist (it was not the case); after the occurrence it does exist (it is the case and it will be the case for ever). Facta are subject to *occurrence laws*. These laws allow or prohibit sequences of transitions. For example, some time after the creation of the factum "loan L has been started", the transition "loan L has been ended" might occur, and in between several other facta may have been created, like "the fine for loan L has been paid". Facta therefore can best be conceived as status changes of a concept of some type (or an object in some class). They appear almost always to be unary concept types. Actually, we will consider in WOSL only unary factum types. This can always be achieved through the conception of new statum types (like loan) as the replacement of an aggregation (cf. Fig. 7). This operation is commonly known as entification or objectification [7].

2.3 World Ontology

We are now able to provide a precise definition of the ontology of a world: a *world ontology* consists of the specification of the state space and the transition space of that world. By the *state space* is understood the set of allowed or lawful states. It is specified by means of the state base and the existence laws. The *state base* is the set of statum types of which instances can exist in a state of the world. The *existence laws* determine the inclusion or exclusion of the coexistence of stata. By the *transition space* is understood the set of allowed or lawful sequences of transitions. It is specified by the transition base and the occurrence laws. The *transition base* is the set of factum types of which instances may occur in the world. Every such instance has a time stamp, which is the event time. The *occurrence laws* determine the order in time in which facta are allowed to occur.

As said before, our notion of world ontology is only a part of the notion of system ontology as proposed in [6]. It exceeds however the common notion, as put forward by [8, 14]. Whereas languages like OWL only regard the state space of a world, WOSL also covers the transition space.

3 The Grammar of WOSL

WOSL is a language for the specification of the ontology of a world. In this section, the grammar of WOSL will be presented. Although it could be fully expressed in modal logic, it seems more appropriate for the practical use of the language to apply a graphical notation. Because of the similarity (but not identity!) between the ontology of a world and the conceptual schema of a database, we have adopted the graphical notation that is applied by one of the fact oriented conceptual modeling languages,

[4] The word "factum" is a Latin word, derived from the transitive verb "facere" of which the meaning is "to make", " to bring about".

namely ORM [7], as the basis. In order to keep the specification of the grammar of WOSL orderly and concise, we present it in a number of figures, exhibited hereafter.

Fig. 3 exhibits the ways in which statum types can be declared. By the *declaration* of a statum type is understood stating that the statum type belongs to the state base of the world under consideration. Statum types can be declared intensionally or extensionally. By intensional we mean the notation of the statum type as a unary, binary, ternary etc. concept type. Intensional notations are referred to be a bold small letter (or a string of small letters). Extensional notations are referred to by a capital letter (or a string of capital letters).

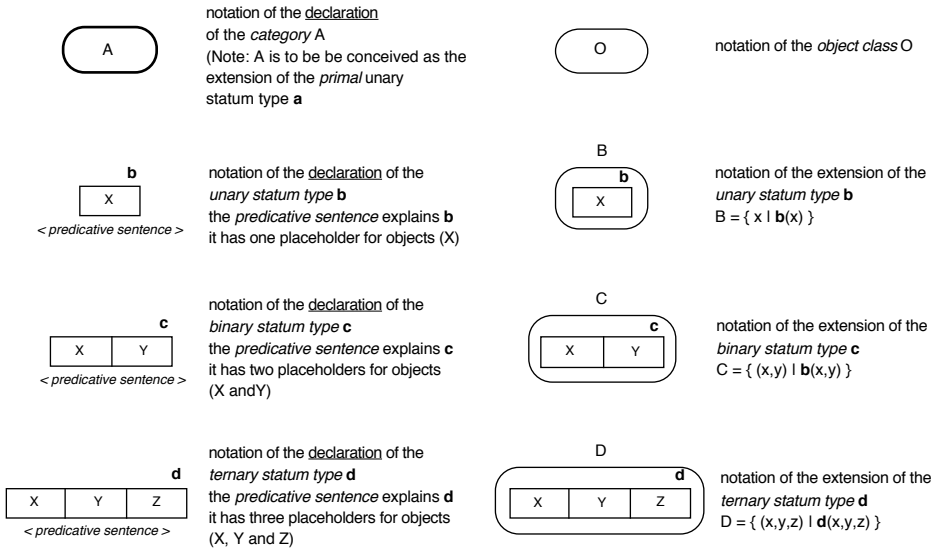


Fig. 3. Statum type declarations

The relationship between the intensional notation of a statum type "**a**" and its extensional notation "A" is: A = {x| **a**(x)}. A category is a primal class. In the ontology of every world there is at least one category. All other classes are the extension of a statum type that is defined on the basis of one or more other classes, including categories by means of reference laws (cf. Fig. 4).

Given the state base of a world, i.e., the statum types that are necessary and sufficient for describing the states of the world, the state space is defined by adding the existence laws. Laws that require the coexistence of objects are presented in Fig. 4. Laws that prohibit the coexistence of objects are presented in Fig. 5. The laws as presented in these figures are the most common ones. Additional, special laws are possible; generally they cannot be easily expressed in a diagram.

On the basis of the declared categories and statum types, new statum types may be defined as so-called derived statum types. Four derivation kinds are distinguished: partition, aggregation, specialization, and generalization. This set of distinct kinds is not exhaustive; they are just the most well known. They are exhibited in Fig. 6 and

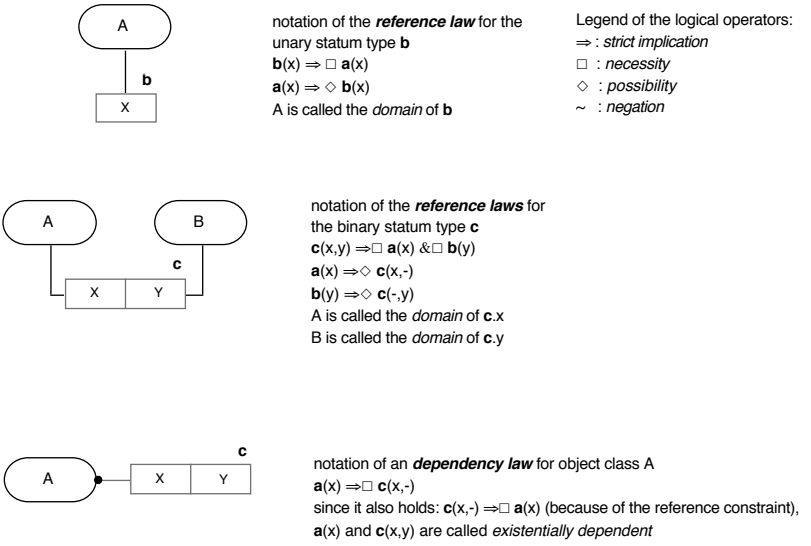


Fig. 4. Coexistence inclusions

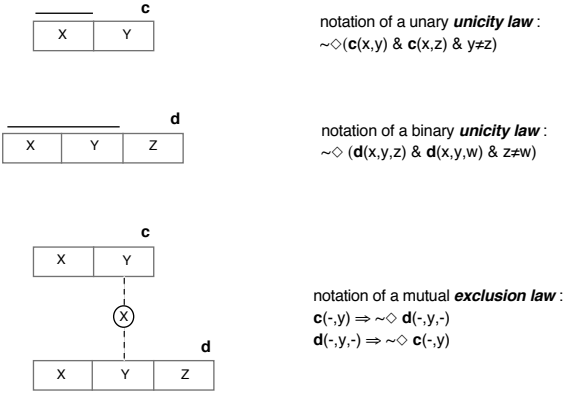


Fig. 5. Coexistence exclusions

Fig. 7. By *partition* is understood the extensional definition of a statum type on the basis of one or more roles of another statum type. If all roles are taken (which requires that there is no special unicity law) we speak of *aggregation*; it is equivalent to the notion of objectification in ORM [9].

Specialization and generalization are often considered to be each other's inverse. This is not correct however. A *specialization* type is always ultimately a subtype of a category (or of a class that is the generalization of a two or more categories). An example of a specialization is STUDENT; it is a subtype of PERSON. Contrary to this, a *generalization* type is always ultimately the union of two or more categories. An example of a generalization is VEHICLE, defined as the union of CAR, AIRCRAFT, and SHIP.

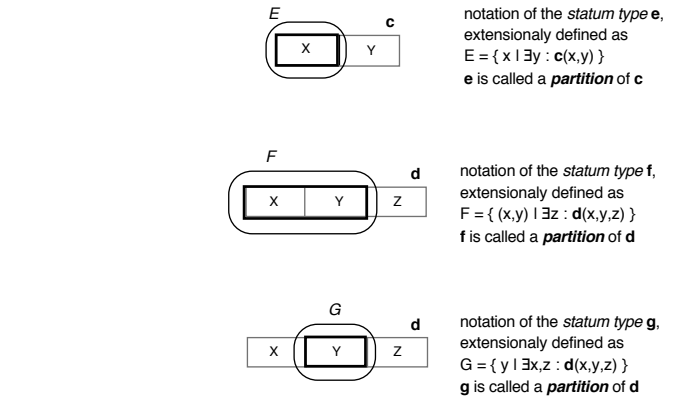


Fig. 6. Deriving statum types as partitions

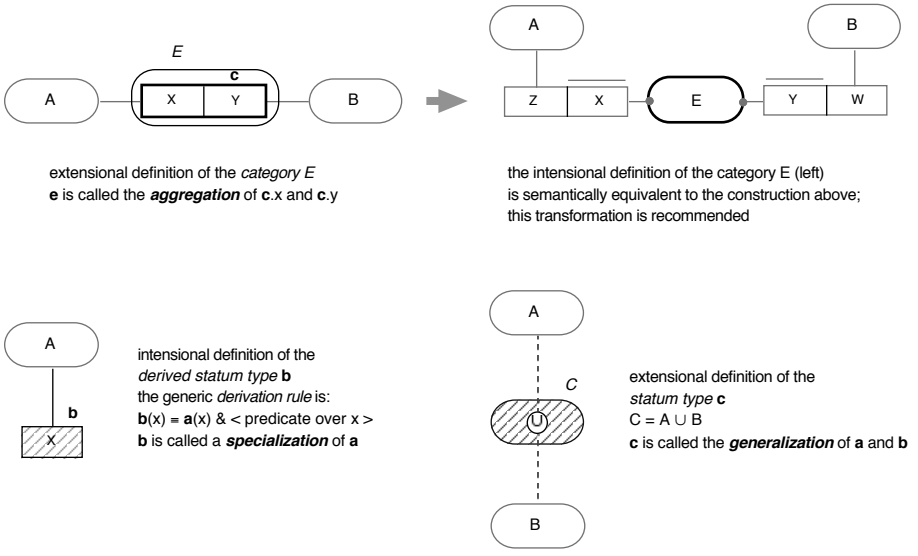


Fig. 7. Aggregation, specialization and generalization

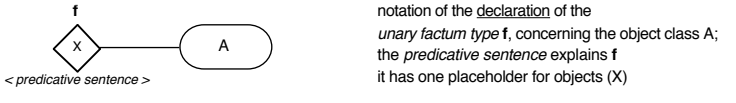


Fig. 8. Declaration of a factum type

Lastly, Fig. 8 exhibits the declaration of factum types. Regarding the occurrence laws, two kinds are distinguished: prerequisite laws and preclusion laws. A *prerequisite law* expresses that a transition must have occurred before some other transition. A *preclusion law* prohibits that a particular transition occurs after some (other) transition has occurred.

4 An Example of the Application of WOSL

In this section, part of the ontology of a library is taken as an example. It is the same case as described in [5]. We assume that the reader is familiar with the operations of a library. Fig. 9 exhibits the ontological model of the library, expressed in WOSL. We assume that the reader has an intuitive understanding of the meanings of the factum types (represented by diamonds). Note that apparently memberships cannot be resigned; this was an intentional part of the case. In Table 1 the applicable occurrence rules for the factum types are listed, which can straightforwardly be derived from the case description. In conformity with [3], the factum types are called event types. Apparently, there are no preclusion laws.

As an illustration of the important difference between stata and facta, let us have a look at the category LOAN and its associated statum types and factum or event types. A loan is fully defined by the membership to which it belongs and the book copy that

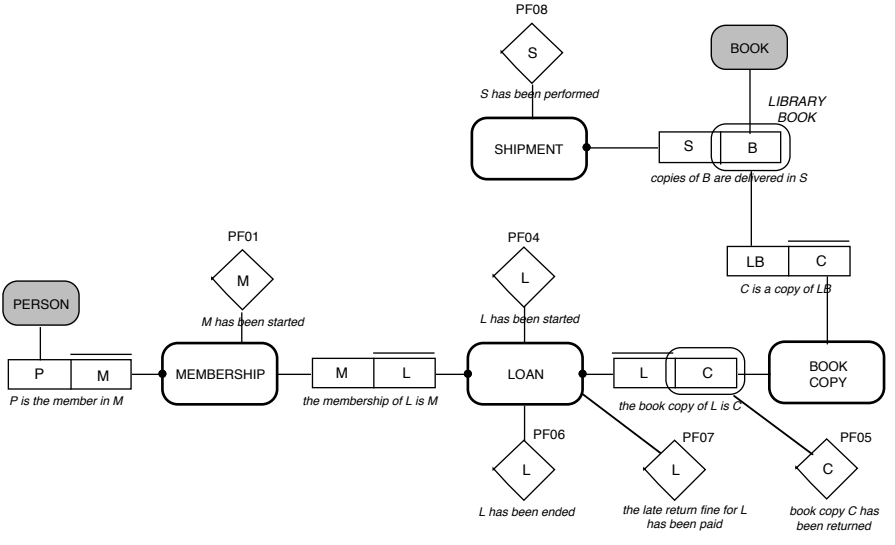


Fig. 9. Ontology of the Library

Table 1. Occurrence laws for the Library

event type	prerequires	precludes
PF01(M)		
PF04(L)	PF01(M): membership(L)=M	
PF05(C)	PF04(L): book_copy(L)=C	
PF06(L)	PF05(C): loan(C)=L	
PF06(L)	PF07(L) <if any>	
PF07(L)	PF05(C): loan(C)=L	
PF08(S)		

it concerns (Note that a loan in this case concerns only one book copy). These two fact types are clearly statum types, they are inherent properties. A loan comes into real existence by an event of the type PF04. During its lifetime, that is ended by a PF06, a PF05 will occur, and possibly also a PF07 (this depends of course on whether the book copy is returned late or not).

5 Conclusions

One of the outcomes of the discussion in section 2 is that in an ontology (or ontological model), names of things are irrelevant, because they are not a property but an attribute. Of course, when one wants to communicate about an object it has to be denoted, but is does not make a difference by which sign or name this is done. This observation clarifies the distinction between an ontology and a conceptual schema, in which usually names are modeled as attributes. Only a meticulous study of the relevant notions regarding factual knowledge, as exercised in section 2, can lead to a clear definition of the notion of the ontology of a world.

Contrary to most ontology languages, the specification of the transition base of a world is included in the proposed language WOSL. This justifies the proposition of WOSL next to the existing ones, because it is not just another ontological language. On the basis of the distinction between stata and facta, a profound understanding has been developed of the state of a world and of transitions (or events) in this world. This distinction is illustrated by the ontological model of the library in section 4. Languages like OWL [14] consider only objects and relationships between objects. They do not address transitions or events.

The difference between a WOSL model and a conceptual schema is twofold. First, the naming of objects is excluded, as this is considered irrelevant on the ontological level. Second, the process or transition perspective is included, such that it is possible to mark the start and the end of the existence of objects. An interesting topic for future research is the exact relationship between WOSL and ORM. For example, can WOSL be considered as an extension of ORM? Next, when regarding the development process of an information system, would it be advisable to first model the ontology of the world under consideration and after that the conceptual schema? Lastly, could the addition of the transition space model be helpful in the design of the information system?

References

1. Berners-Lee, T., J. Hendler, O. Lasilla, The Semantic Web, *Scientific American*, May 2001.
2. Bunge, M.A.: *Treatise on Basic Philosophy, vol.4, A World of Systems* (D. Reidel Publishing Company, Dordrecht, The Netherlands 1979
3. Dietz, J.L.G., The Atoms, Molecules and Fibers of Organizations, *Data and Knowledge Engineering*, vol. 47, pp 301-325, 2003
4. Dietz, J.L.G. and N. Habing. A meta Ontology for Organizations. In *Workshop on Modeling Inter-Organizational Systems (MIOS), LNCS 3292*. 2004. Larnaca, Cyprus: Springer Verlag.

5. Dietz, J.L.G., T.A. Halpin, Using DEMO and ORM in concert – A Case Study, in: Siau, K. (ed), *Advanced Topics in Database Research*, vol. 3, IDEA Publishing, London, 2004
6. Dietz J.L.G., *Enterprise Ontology – theory and methodology*. Springer-Verlag Heidelberg Berlin New York (forthcoming)
7. Falkenberg, E.D., (ed.), *A Framework for Information Systems Concepts*, IFIP 1998 (available as web edition from www.wi.leidenuniv.nl/~verrynst/frisco.html)
8. Gruber T., (1995), Towards Principles for the Design of Ontologies Used for Knowledge Sharing, International Journal of Human-Computer studies, 43 (5/6): 907 – 928.
9. Halpin, T.A., *Information Modeling and Relational Databases*. San Francisco: Morgan Kaufmann, 2001
10. Morris, C.W., *Signs, Language and Behavior*, New York, 1955.
11. Searle, J.R., *The Construction of Social Reality*, Allen Lane, The Penguin Press, London, 1995
12. Sowa, J.F., *Conceptual structures: information processing in mind and machine*, Adisson-Wesley P.C., 1984
13. Ross, D.T., *Plex 1: sameness and te need for rigor*, and *Plex 2: sameness and type*. Softech Inc., Waltham, MA, 1975
14. W3C, *OWL, Web Ontology Language Overview*, http://www.w3.org/TR/2004/REC-owl-features-20040210/

Using ORM to Model Web Systems

Olga De Troyer, Sven Casteleyn, and Peter Plessers

Vrije Universiteit Brussel, WISE, Pleinlaan 2, 1050 Brussel, Belgium
{Olga.DeTroyer, Sven.Casteleyn, Peter.Plessers}@vub.ac.be
http://wise.vub.ac.be

Abstract. In this paper, we describe how ORM is extended, and combined with Concurrent Task Trees (CTT) to model the content as well as the functionality of a web system in the web design method WSDM. As WSDM uses an audience driven design approach, the use of ORM is somewhat different from its use for data modeling in the context of databases. We discuss the differences. We also discuss the benefits of using ORM for our purpose, modeling web systems using an audience driven approach.

1 Introduction

In the last years, web sites have evolved from a simple collection of hypertext pages towards large web applications supporting both static and dynamic content and complex (business) processes. Although it is still easy to publish a couple of pages, more and more it is recognized that appropriate design methods are needed to develop large and complex web sites. In the past, web sites were created opportunistically without any regard for methodology. The increased complexity soon lead to usability problems with sites developed in an ad-hoc manner [1]: the information is badly organized, the navigation structure is non-transparent and difficult to grasp for visitors and the look & feel may be inconsistent. Consequently, users fail to make a comprehensive mental model of the website. Furthermore, maintainability, evolution and extensibility of these web sites are highly problematic. Therefore, in the past, a number of web design methods have been proposed. Examples include WSDM [2], WebML [3], OOHDM [4], OO-H [5].

Most of the existing web design methods recognize that in order to design a web system[1] due consideration should be given to the following aspects: content, functionality, navigation and presentation. To model the content of a web system, the methods usually rely on an existing data modeling technique, e.g., WebML uses the ER model, OOHDM as well as OO-H applies the class diagrams of UML. Recent methods like HERA [6] uses semantic web technology (e.g., RDF (S)). As far as we are aware, WSDM is the only web design method that uses ORM to model the content of a web system. To model functionality, WebML provides operation units that can be associated to data entry units to process information entered by a user, and predefined units for the standard data management functionality. Also a workflow data model (composed of processes, activities, etc) can be defined. OOHDM uses UML's user interaction diagrams to model the interaction with the user. In Hera, the

[1] We will use the term web system to indicate web sites as well as web applications.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 700 – 709, 2005.
© Springer-Verlag Berlin Heidelberg 2005

modeling of user interaction is based on forms and queries. Forms are used to retrieve information from the user and are based on the XForms standard; queries are used to update and retrieve data and are expressed in SeRQL. To model functionality, WSDM combines an extended form of ORM with Concurrent Task Trees [7] (CTT).

The purpose of this paper is to explain how ORM is used in WSDM to model the content of the web system and how it has been extended and combined with CTT to allow for the modeling of web functionality. We suppose that the reader is familiar with ORM. The rest of the paper is organized as follows. Section 2 provides an overview of the different phases of WSDM. Section 3 elaborates on the use of ORM to model content and functionality. In Section 4 we discuss the differences with ORM as data modeling method for databases and provide conclusions.

2 WSDM Overview

The web site design method WSDM follows the *audience driven* design philosophy. This means that the different target audiences and their requirements are taken as starting point for the design and that the main structure of the web system is derived from this. Concretely, this will result in different navigation path (called tracks) for different kinds of visitors in the organizational structure of the web system.

WSDM is composed of a number of phases. In the first phase the mission statement for the web system is formulated. The goal is to identify the purpose of the web system, as well as the subject and the target users. The *mission statement* is formulated in natural language and must contain the elements mentioned. The mission statement establishes the borders of the design process. It allows (in the following phases) to decide what information or functionality to include or not, how to structure it and how to present it.

The next phase is the 'Audience Modeling' phase. The target users identified in the mission statement should be refined into so-called *audience classes*. The different types of users are identified and classified into audience classes. How this is done can be find in [8]. The classification is based on the requirements (information-, as well as functional-, and usability requirements) of the different users. Users with the same information and functional requirements become members of the same audience class. Users with additional requirements form audience subclasses. In this way a hierarchy of audience classes is constructed. For each audience class relevant characteristics (e.g., age, experience level) are given.

The next phase, the 'Conceptual Design' is used to specify the content, functionality and structure of the web system at a conceptual level. A conceptual design makes an abstraction from any implementation or target platform. The content and functionality are defined during the 'Task Modeling' sub phase. This will be described in detail in the next section. The overall conceptual structure including the navigational possibilities for each audience class is defined during the 'Navigational Design' sub phase. We will not elaborate on this sub phase.

During the 'Implementation Design' phase, the conceptual design models are completed with information required for the actual implementation. It consists of three sub phases: 'Site Structure Design', 'Presentation Design' and 'Data Source Mapping'. During Site Structure Design, the conceptual structure of the web system is

mapped onto pages, i.e. it is decided which components (representing information and functionality) and links will be grouped onto web pages. For the same Conceptual Design, different site structures can be defined, targeting different devices, contexts or platforms. The Presentation Design is used to define the look and feel of the web system as well as the layout of the pages. The 'Data Source Mapping' is only needed for data-intensive web sites. In case the data will be maintained in a database, a database schema is constructed (or an existing one can be used) and the mapping between the conceptual data model and the actual data source is created.

The last phase is the 'Implementation'. The actual implementation can be generated automatically from the information collected during the previous phases. As proof-of-concept, a transformation pipeline (using XSLT) has been implemented, which takes as input the different models and generates the actual implementation for the chosen platform and implementation language. This transformation is performed fully automatically. An overview of this transformation pipeline is given in [9].

3 Modeling Content and Functionality in WSDM

During Audience Modeling, the information-, functional-, and usability requirements of the potential visitors are identified and different Audience Classes are defined. The goal of the Conceptual Design is to turn these (informal) requirements into high level, formal descriptions which can be used later on to generate (automatically) the web system.

During conceptual design, we concentrate on the **conceptual** "what and how" rather than on the visual "what and how". The conceptual "what" (content and functionality) is covered by the Task Modeling step, the conceptual "how" (conceptual structure and navigation possibilities) by the Navigational Design.

Instead of starting with an overall conceptual data model, like most web design methods do, WSDM starts with analyzing the requirements of the different audience classes. This will result in a number of tiny conceptual schemas, called *Object Chunks*. Later on these conceptual schemas are integrated into an overall conceptual data model [10]. This approach has several advantages:

- It allows the developer concentrating on the needs of the actual users rather than on the information (and functionality) already available in the organization (which is not necessarily the information needed by the users. In addition the way the information is organized and structured in the organization is not necessarily the way external users need it).
- It gives consideration to the fact that different types of users may have different requirements: different information or functional requirements; but they may also require a different structure or terminology for the information. By modeling the requirements for each audience class separately we can give due consideration to this. E.g., we can avoid that the user is overloaded with information of which most is not relevant to him.

Analyzing and modeling the information and functionality needed for the different audience classes is done in the Task Modeling phase.

3.1 Task Modeling

The tasks the members of an audience class need to be able to perform are based on their requirements (informally) formulated during audience classification. To come to detailed task models, a task analysis is done. I.e. for each information and functional requirement formulated for an audience class, a task is defined. Each task is modeled into more details using an adapted version of the task modeling technique CTT [11] (called CTT+). This has been described into more detail in [7].

CTT was developed in the context of Human-Computer Interaction to describe user activities. CTT looks like hierarchical task decomposition but it distinguishes four different categories of tasks (user tasks, application tasks, interaction tasks, and abstract tasks) and it also allows expressing temporal relationships among tasks. In addition, CTT has an easy to grasp graphical notation. For its use in WSDM, we slightly adopted the technique to better satisfy the particularities of the Web:

1. WSDM do not consider user tasks. User tasks are tasks performed by the user without using the application (such as thinking on or deciding about a strategy). They are not useful to consider at this stage of the design. We only use:
   - *Application tasks*: tasks executed by the application. Application tasks can supply information to the user, perform some calculations or updates.
   - *Interaction tasks*: tasks performed by the user by interaction with the system.
   - *Abstract tasks*: tasks that consists of complex activities, and thus requiring decomposition into sub tasks.
2. A (complex) task is decomposed into (sub)tasks. Tasks are identified by means of their name. The same task can be used in different sub-trees, and a task may be re-used inside one of its sub-trees (expressing recursion). CTT prescribes that if the children of a task are of different categories then the parent task must be an abstract task. WSDM do not follow this rule. We use the category of the task to explicitly indicate who will be in charge of performing the task. The convention is that for an interaction task, the user will be in charge; for an application task the application will be in charge.
3. CTT has a number of operators (with corresponding graphical notation) to express temporal relations among tasks. E.g., tasks can be *order independent, concurrent* (with or without information exchange); tasks may exclude each other (*choice*); one task may *enable* (with or without information passing) or *deactivate* another task; a task can be *suspended* to perform another task and *resumed* when this last task is completed; a task can be *iterated* (a defined or undefined number of times); and a task can be *optional* or mandatory. An extra operator for dealing with *transactions* has been added.

We illustrate this task modeling process by means of an example. The example is taken from the Conference Review System case [12], which aimed at providing a web system for supporting the paper review process of a conference. We start with the following requirement formulated for the 'Author' Audience Class:

"*An author must be able to consult the information about its submissions*"

For this requirement, the following (abstract) task is defined: "*Find Information about Author's Submissions"*. This task can be decomposed into two more elementary tasks (see figure 1). The two subtasks are informally described as follows:

1. *Find Author's Submissions*: Give paper-id and title of all papers submitted by the author as main author or as co-author and allow the user to select a paper. This is an interaction task because the task requires interaction with the user.
2. *Give Submission's Information*: For a selected paper give paper-id, title, abstract, main author (name) and co-authors (names), track and if available the subjects, and file (url) containing the paper. This is an application type of task.

The temporal relationship expressed between the two sub tasks (symbol []>> in figure 1) expresses that finishing the task *Find Author's Submissions* enables the task *Give Submission's Information* and some information must be passed from the first to the second task (i.e. the selected submission).

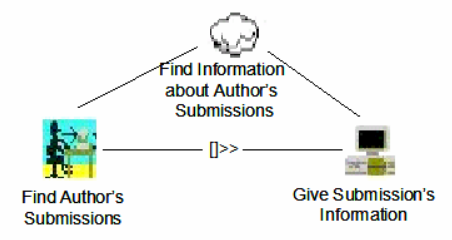


Fig. 1. Task Model for the task "Find Information about Author's Submissions"

The task models created in this way allow for a first level of description of the functionality to be provided by the web system (i.e. they describe a kind of workflow). The elementary tasks will be further specified by creating object chunks, which are tiny conceptual schemas (see next section). In the following sub phase the task models and the temporal relationships are used to derive the navigational model, which expresses the navigational possibilities for the different audience classes.

3.2 Detailed Information and Functional Modeling

When a task model is completed, for each elementary task an object model, called *Object Chunk*, is created that (formally) models the necessary information and functionality needed for this elementary task. For this an extended version of ORM is used. Figure 2 shows the Object Chunk for the task *Find Author's Submissions* of Figure 1. We will explain it later on.

The main purpose of an Object Chunk is to represent the information needed by the user when he has to perform the associated task. If the requirement is a pure information requirement (i.e. the user is only looking for information; he doesn't need to perform actions) then an Object Chunk can be considered as a conceptual description of the information that will be displayed on (a part of) the screen. For this purpose standard ORM is sufficient. However, to be able to deal with functionality (e.g., filling in a form), it was necessary to extend ORM:

1. To express functionality we need to be able to refer to instances of Object Types. For this we use *referents.* Graphically, a referent is placed inside the circle depicting its Object Type. For Value Types, instances (and therefore also referents)

are values, e.g., 'Casteleyn' is an instance of the *PersonName,* 1 is an instance of *Rate*. A *generic marker* is used to represent an instance of an Entity Type, e.g., **a* in figure 2 represents an instance of type *Author*. Sets of referents are represented using the traditional set brackets '{' and '}', e.g., {*p} placed in the Object Type *Paper* would represents a set of Paper instances. The traditional set operators (union, intersection, etc.) can be used on sets. The notation {…}@ indicates the cardinality of the set.

To represent a relationship between instances, we can place the referents either in the Object Types (if no confusion is possible) or in the boxes of the roles of the relationship.

2. To allow communication between tasks, parameters (input- as well as output parameters) are specified for Object Chunks. E.g., the Object Chunk *AuthorSubmission* (figure 2) has an instance of type Author as input parameter (represented by the referent *a) and an instance of type Paper as output parameter (represented by the referent *p). Input parameters usually restrict the information that should be presented to the user. E.g., the input parameter *a of type Author, is used to express that only the information related to this particular author should be shown. This is indicated by putting this parameter in the corresponding Object Type. In this way only the Paper-instances 'with main' Author equal to *a, and the Paper-instances 'with co' Author equal to *a will be considered (and displayed on the corresponding page of the actual web system). In fact, the use of the parameter *a in the Object Type Author restricts the complete schema to a kind of view. The fact that for the Object Type Paper two Value Types (PaperTitle and PaperId) are included indicates that the relevant Paper-instances should be shown by means of their paper title and paper id.
3. To model that some information must be selectable (e.g., papers should be selectable to ask for the details of a paper; or when filling in the review form, it

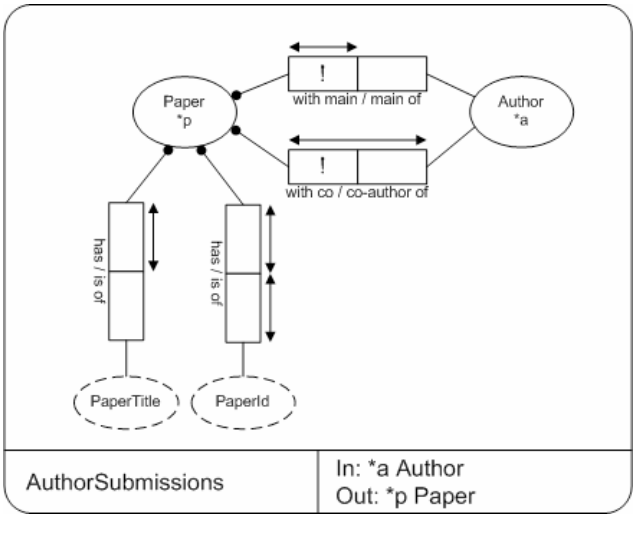


Fig. 2. Object Chunk *AuthorSubmissions*

must be possible to select the rate from a given list) a selection operator is used, represented by the symbol '!'. E.g., !{*p} expresses that the user can select an instance from the set {*p}. The symbol '!!' is used to indicate that more than one instance is selectable.

The selection symbol is placed inside the circle depicting an Object Type when an instance can be selected from the (possibly restricted) population of this Object Type. Similar, to indicate that it is possible to select an instance from the (possibly restricted) population of an Object Type role, the selection symbol is placed inside the box of this role. E.g., in figure 2, the '!' in the roles 'with main' and 'with co' indicates that the user is able to select a Paper-instance from the set of Paper-instances 'with main' Author equal to *a and from the set of Paper-instances 'with co' Author equal to *a. Intuitively, this means that the user can select a paper from all papers submitted by the author *a either as main author or as co-author.

4. To express interactive input (e.g., needed to fill in a form) a principle similar to that of the selection is used. The symbol '?' is used for the input of a single value, '??' is used for expressing interactive input of more then one value. Note that these symbols can only be applied to Value Types. Entity Type instances cannot be entered directly. They should be created by the system using the NEW operator (see further on).
5. To assign values to referents the assignment operator ('=') is used, e.g., in figure 3, which shows the Object Chunk for a task *Register New Paper*. **t = ?* in the Value Type PaperTitle indicates that the user should enter the paper title, which is then assigned to the referent *t.

The value of a referent can be changed by means of the operator '→'. *t → ? in the Value Type PaperTitle would allow the user to change the given paper title (represented by *t) by entering a new title; {*a} → ∅ will replace the value of the referent set {*a} by the empty set.

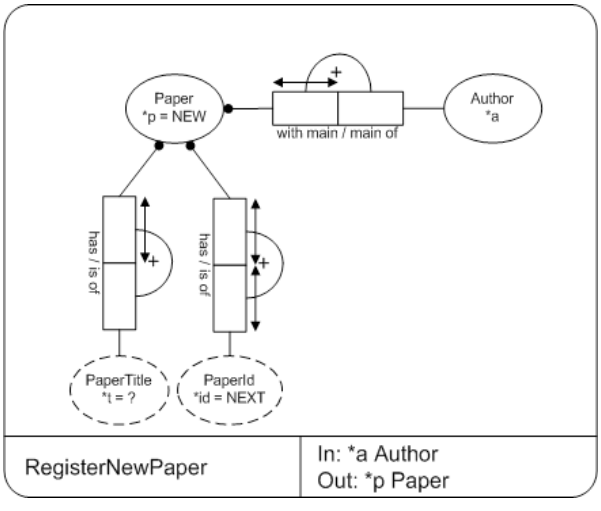


Fig. 3. Object Chunk *RegisterNewPaper*

6. To manipulate the data itself, a number of primitives are available:
   - 'NEW' indicates the creation of a new Object Type instance. E.g., in figure 3, *p = NEW in the Object Type Paper indicates the creation of a new Paper instance, which is assigned to the referent *p;
   - 'REMOVE' is used to indicate the removal of one or more instances from an Object Type;
   - '+' above a relation indicates the addition of a relationship. In figure 3, these are used to specify that the relationships (*p, *a), (*p, *t) and (*p, *id) should be added.
   - '-'above a relation indicates the removal of a relationship;
7. Furthermore, we have:
   - '==' for testing equality of values;
   - 'IS' for testing membership of an Object Type;
   - 'EXIST' to test the existence of an instance;
   - 'NEXT' to generate a unique system identifier (used in figure 3 to generate the following paper id);
   - 'TODAY' represents the current date;
   - 'UPLOAD' and 'EMAIL' are supposed to be built-in functions.

3.3 The Business Information Model

The disadvantage of modeling the requirements of the different audience classes by means of separated Object Chunks is that they need to be related to each other and possibly also integrated. Indeed, different chunks may deal with the same information. For example, Authors as well as PC-Members need information about papers. This means that different Object Chunks may model (partially) the same information. This redundancy is deliberately (the same information may need different representations or different navigation paths for different audience classes) and will not cause any problems as long as we recognize that it is indeed the same information and that it is maintained in a single place. Therefore, the different chunks are related to each other. In addition, an integrated schema may be useful if one wishes to maintain the information in a database. In that case, such an integrated schema, called *Business Information Model*, will be the conceptual schema of this database. It is possible that the web system will use an existing database. In this case the different chunks need to be defined as views on the schema of this database. How this should be done is outside the scope of this paper. In [13] we have proposed an approach to link (already during modeling) the concepts used in the different Object Chunks to an ontology. In this way the integration can be done in an automatic way and it has the additional advantage that it paves the way for semantic annotations.

The mapping of the Object Chunks (or the Business Information Model) onto the actual data source(s) is considered in the Data Source Mapping step of the Implementation Design.

4 Discussion and Conclusions

In this paper, we have presented how ORM is used in the web design method WSDM to model the content and the functionality of a web system. For this purpose, ORM

was extended with referents to be able to refer to instances, and with a number of operators for modeling selection of information, interactive input and data manipulation. Workflow is captured by means of task models, expressed in an extended version of CTT. The task models and the ORM descriptions are linked by defining an Object Chunk for each elementary task in a task model.

Next to the capability to define functionality, there are a number of fundamental differences in how ORM is used in WSDM, compared with the use of ORM as data modeling technique for databases:

- ORM is used to model the information requirements of a single task. This results in tiny conceptual schemas, called Object Chunks, comparable to views in databases. Later on the different Object Chunks can be integrated into an overall data model.
- When using ORM for data modeling, constraints are specified to allow checking the integrity of the database and are valid for the entire model. This is not the case for our Object Chunks. Because chunks are used to model requirements in the context of certain task and for a particular audience class, the constraints in a chunk only express constraints formulated or applicable in that context. These are not necessarily the constraints that apply in general for the information considered. E.g., in general a paper has different reviews, but for the task of reviewing papers for the audience class PC Member, a paper only has one review.

In the early days of WSDM, class diagrams were used to model content [2]. However, very quickly it turns out that these diagrams were not suitable for the use in Object Chunks. People always tended to put all possible attributes in the classes, even if there were not needed for the purpose of the task at hand. With ORM, in which all relationships are modeled at the same level, we didn't have this problem. Also, the integration of the Object Chunks turned out to be simpler with ORM.

References

1. Nielsen, J.: Designing Web Usability: The Practice of Simplicity. New Riders Publishing (2000)
2. De Troyer, O., Leune, C.: WSDM: A User-Centered Design Method for Web Sites. In Computer Networks and ISDN systems, Proceedings of the 7th International World Wide Web Conference, Publ. Elsevier, Brisbane, Australia (1998) 85-94
3. Ceri, S., Fraternali, P., Bongio, A.: Web Modeling Language (WebML) : a modeling language for designing Web sites. In Proceedings of the ninth World Wide Web Conference (2000)
4. Schwabe, D., Rossi, G., Barbosa, S.: Systematic Hypermedia Application Design with OOHDM. In Proceedings of ACM Hypertext'96 Conference (1996) 116-128
5. Gómez, J., Cachero, C., Pastor, O.: Conceptual Modelling of Device-Independent Web Applications. IEEE Multimedia Special Issue on Web Engineering (2001) 26-39
6. Vdovjak, R., Frasincar, F., Houben, G.J., Barna, P.: Engineering Semantic Web Information Systems in Hera. Journal of Web Engineering (JWE), Vol. 2, Nr. 1-2, Rinton Press (2003) 3-26

7. De Troyer, O., Casteleyn, S.: Modeling Complex Processes for Web Applications using WSDM. In Proceedings of the Third International Workshop on Web-Oriented Software Technologies (held in conjunction with ICWE2003), IWWOST2003 (also on http://www.dsic.upv.es/~west/iwwost03/articles.htm), Eds. Daniel Schwabe, Oscar Pastor, Gustavo Rossi, Luis Olsina, Oviedo, Spain (2003)
8. Casteleyn, S., De Troyer, O.: Structuring Web Sites Using Audience Class Hierarchies. In Conceptual Modeling for New Information Systems Technologies, ER 2001 Workshops, HUMACS, DASWIS, ECOMO, and DAMA, Lecture Notes in Computer Science , Vol. 2465, Publ. Springer-Verlag, ISBN 3-540-44-122-0, Yokohama, Japan (2001)
9. Plessers, P., Casteleyn, S., Yesilada, Y., De Troyer, O., Stevens, R., Harper, S., Goble, C.: Accessibility: A Web Engineering Approach. In Proceedings of the 14th International World Wide Web Conference (WWW2005), Chiba, Japan (2005)
10. De Troyer, O., Plessers, P., Casteleyn, S.: Conceptual View Integration for Audience Driven Web Design. In CD-ROM Proceedings of the WWW2003 Conference, IW3C2 (also http://www2003.org/cdrom/html/poster/), Budapest, Hongary (2003)
11. Paterno, F.: Model-Based Design and Evaluation of Interactive Applications. Eds. Ray, P., Thomas, P., Kuljis, J., Springer-Verlag Londen Berlin Heidelberg, ISBN 1-85233-155-0, 2000
12. De Troyer, O., Casteleyn, S.: The Conference Review System with WSDM. In First International Workshop on Web-Oriented Software Technology, IWWOST'01, (also http://www.dsic.upv.es/~west2001/iwwost01/), Eds. Oscar Pastor, Valencia (University of Technology), Spain (2001)
13. De Troyer, O., Plessers, P., Casteleyn, S.: Solving Semantic Conflicts in Adience Driven Web Design. In Proceedings of the WWW/Internet 2003 Conference (ICWI 2003), Vol. I, Eds. Pedro Isaías and Nitya Karmakar, Publ. IADIS Press, ISBN 972-98947-1-X, Algarve, Portugal (2003) 443-450

Object Role Modelling for Ontology Engineering in the DOGMA Framework

Peter Spyns

Vrije Universiteit Brussel, STAR Lab, Pleinlaan 2, Gebouw G-10, B-1050 Brussel, Belgium
Peter.Spyns@vub.ac.be
http://www.starlab.vub.ac.be

Abstract. A recent evolution in the areas of artificial intelligence, database semantics and information systems is the advent of the Semantic Web that requires software agents and web services exchanging meaningful and unambiguous messages. A prerequisite for this kind of interoperability is the usage of an ontology. Currently, not many ontology engineering methodologies exist. This paper describes some basic issues to be taken into account when using the ORM methodology for ontology engineering from the DOGMA ontology framework point of view.

1 Introduction

A recent evolution in the areas of artificial intelligence, database semantics and information systems is the advent of the Semantic Web. An essential condition to the actual realisation of the Semantic Web is semantic interoperability, which is currently still lacking to a large extent. Nowadays, a formal representation of a (partial) intensional definition of a conceptualisation of an application domain is called an ontology [11]. The latter is understood as a first order vocabulary with semantically precise and formally defined logical terms that stand for concepts and their inter-relationships of an application domain. This paper presents some basic thoughts on how to adapt an existing conceptual schema modelling methodology called Object Role Modelling (ORM [12]) [1] for ontology engineering within the DOGMA initiative framework. The paper summarises and extends previous work at VUB STAR Lab.

Early work on combining DB modelling with insights from linguistics is [32]. A more recent overview can be found in [21]. In the ontology engineering community, the importance of grounding the logical terms seems an issue rather neglected. Exceptions are researchers active at the intersection of natural language processing and ontology engineering, e.g. [4,15,22].

The paper is organised as follows. The following section (2) shortly explains how ontologies compare to conceptual data models. Subsequently section 3 presents the difference between formal and natural interpretation. Section 4 introduces the DOGMA ontology framework. In section 5, the distinctions between ORM and DOM^2 are being discussed, ORM being a conceptual data modelling methodology while DOM^2 is the DOGMA ontology modelling methodology. Section 6 contains the discussion, followed section 7 that ends the paper by outlining the future work and by giving some concluding remarks.

[1] We assume that the reader is familiar with ORM.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 710 – 719, 2005.
© Springer-Verlag Berlin Heidelberg 2005

2 Ontologies vs. Data Models

A data model is, in principle, "tuned" towards a specific application, and therefore has less or no needs for explicit semantics (since sharing is not required). The conceptual schema vocabulary is basically to be understood intuitively (via the terms used) by human developers. A conceptual schema for an application is a "parsimonious" model, i.e., only the distinctions (between entities, relations and characteristics) relevant for that particular application matter and are thus considered. This also applies to a global schema integrating multiple local schemas [23].

An ontology, at the contrary, is a "fat model" as it is, by definition, to be shared across many applications to support interoperability and, therefore, has to be broader and deeper (necessitating a larger coverage and higher granularity). Most importantly, interoperability requires a precise and formal definition of the intended semantics of an ontology (see also [17,26] for more and other details in this discussion). Alternatively, the difference can be stated in terms of the number of models possibly to comply with: one (data model) versus many (ontology) [3].

Thus, the key feature that distinguishes a conceptual data model from an ontology is the *availability of definitions that unambiguously "fix" the intended semantics (be it only partially) of the conceptual terminology*.

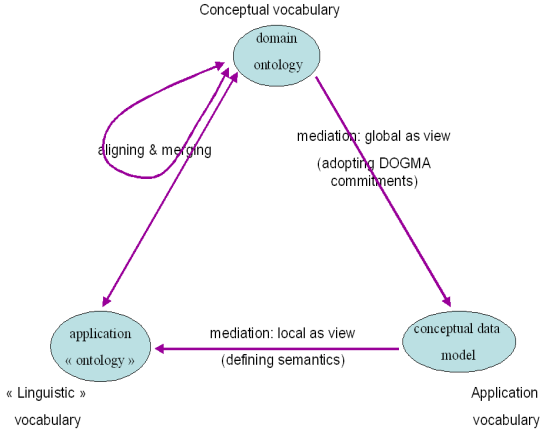


Fig. 1. ontological "triangulation"

Even in the ontology literature, authors do not always make a clear distinction between a global domain concept and a local conceptual schema term (application vocabulary – see Figure 1 right lower corner). In particular, application or local ontologies are a dubious case: labels of the conceptual schema of a database are often "lifted" into concepts, sometimes being replaced by a synonym or orthographical variant for ease of reading and simplicity. However, without an accompanying specification of their meaning (e.g. a gloss and dictionary style definition), a shared and agreed meaning cannot be reached. A mere term cannot suffice as the understanding remains intuitive (i.e., on the linguistic level – see Figure 1 left corner). As a result, the (global) conceptual and (local) language or application levels become quickly mixed up. This can particularly be harmful when aligning and merging ontologies. Only if accompanying glosses or meaning explanations are available for the terms, genuine semantic merging or alignment can be achieved (conceptual vocabulary – see Figure 1 top).

In case one or several database schemas have been designed on basis of an ontology, it would be natural to see that the ontology terms are used inside of the conceptual schema (global ontology [31:p.32]). The inverse scenario is to create an

application or local ontology [11:p.10], by extracting an ontology from a conceptual schema and defining the semantics of its terms that are promoted to concept labels (local as view – e.g., [18]). An application ontology can subsequently be merged [10] with more general domain ontologies and/or other application ontologies[2]. A third scenario is to use a hybrid approach combining global and local ontologies [31:p.32].

Some of these issues are already implicitly touched upon by Halpin (e.g. [12:p.74]), but he doesn't foresee any (i) mechanism to define the vocabulary and he doesn't (ii) go beyond the application at hand and its associated (or exemplary) data population when modelling. In particular, the labels denominating the object types or abstract entities are chosen at the will of the modeller. Halpin himself implicitly proves our point by explaining his usage of 'No.' versus 'Nr.' [12:p.42]. In our framework, an application commits to an ontology by mapping its local terminology to a selection (by the application developer) of well defined concepts (see also section 4). Not every local term may be relevant to be mapped to a global concept.

3 Natural vs. Formal Semantics

Language words have acquired meaning over time (resulting sometimes in synonymy and homonymy), which is recorded in (electronic) dictionaries and thesauri. Formal terms are used in axiomatisations to reason about meaning and compute entailment. However, we concur with Farrugia who states that:

> Model-theoretic semantics does not pretend, and has no way to determine what certain words and statements "really" mean. (…) It [= model theoretic semantics] offers no help in making the connection between the model (the abstract structure) and the real world [8:pp.30-31].

The fundamental problem is that logical theories are "empty", i.e., represent mathematical structures that receive their meaning from an interpretation function that maps the logical vocabulary to (sets of) entities in the universe of discourse [9] (definition by extension). How this mapping exactly happens is open, and many models and interpretations can exist in parallel (possible worlds). In addition, Guarino has shown that an ontology deals with intension rather than with extension [11], so the question remains how the mathematical structure (c.q. ontology) receives its meaning.

Our solution is to use natural language to bridge this gap. Many terminological resources already exist, mainly in technical and/or professional areas – WordNet being an exception with its general language content. A shared understanding and consensuality on meaning necessarily passes through the use of natural language [18], whereby a clear distinction has to be made between the natural language vocabulary of humans and the logical vocabulary of the ontology.

It means that before the formal stage of modelling can start, an informal, but even more important stage has to happen where the relevant stakeholders (i) identify the important domain vocabulary (natural language), (ii) distil the important notions and

[2] One has to beware of "just" shifting the interoperability problem from the schema to the ontology level, thereby replacing schema integration by ontology integration.

relations from the natural language term collection, (iii) select (if available) or compose (otherwise) an informal (but clear) definition that describes as adequately as possible the intended meaning of the notion or relation aimed at and on which all the stakeholders involved agree, (iv) choose an appropriate concept or relation label (the logical term) to represent a definition, and (v) link it to the various domain terms collected that express that notion (synonyms, translations). Care has to be taken to avoid ambiguity on the conceptual level – e.g., the situation or context in which a word (in a particular language) is used should resolve its polysemy [6].

The tasks mentioned above resemble very much the work of terminologists who, at least in our opinion, should become more involved in the ontology engineering process [28]. A genuine ontology engineering methodology should therefore formally include this stage in its flow just as genuine ontology infrastructure should have the necessary software tools and modules to support this. After which, the knowledge engineers can model the domain and/or application axiomatisations using most of the ORM constraints (see also [29]).

4 DOGMA: Developing Ontology Guided Mediation for Agents

4.1 The DOGMA Framework in Short

VUB STAR Lab has its own ontology engineering framework called "Developing Ontology Guided Mediation for Agents" [18]. The original foundations of DOGMA, taking into account database modelling theory and practice [13,19] – in particular ORM [12], had to be refined. Recently, the DOGMA framework has been refined to add the distinction between the language and conceptual levels by formalising the context and introducing language identifiers [6,28]. The DOGMA double articulation[3] decomposes an ontology into an *ontology base* (intuitive binary and plausible conceptualisations) and a separate layer, called *commitment layer*, of instances of explicit ontological commitments by an application (see Figure 2) [13,26].

Figure 2 illustrates that for application vocabulary, through mappings and commitment rules that impose extra constraints (e.g., uniqueness), meta-lexons (see below) are selected from a larger ontology base and as a result commits these local terms to definitions in a concept definition server (not shown).

4.2 The DOGMA Ontology Modelling Methodology (DOM^2) Fundamentals

We propose to organise the ontology modelling process in two major steps: (i) a linguistic step and (ii) a conceptual step. The latter is subdivided in a domain and application axiomatisation phase. Note that DOM^2 still lacks aspects of distributed

[3] The original notion of "double articulation" comes from Martinet [0: pp. 157-158] who explained how humans with a limited set of sounds (first level) are able to form meaningful elements ("subunits" of words) that, in turn, can be combined to create an unlimited number of words expressing ideas (second level). In an analogous way, the ontology base contains concepts and relations (albeit potentially a very large collection) that are combined into meta-lexons (first level), of which particular selections are formally constraint by semantic rules (commitment rule) – e.g., cardinality, mandatoriness, …. – to create an infinite number of interpretation variations (second level) (see figure 2).

collaborative modelling, which is extremely important for reaching a common agreement about meaning. We hope to draw upon existing practices from the terminology community to refine our modelling methodology on these aspects [28].

Linguistic Stage

The starting point is, just as in ORM, the *verbalization* of information examples as *elementary facts*. How to get relevant material and to produce these elementary facts is not discussed here.

The next step consists of transforming these elementary facts into formal *DOGMA lexons*: i.e., a sextuple <(γ,ζ): $term_1$, role, co-role, $term_2$>. Informally we say that a lexon is a binary fact that may hold for some domain, expressing that within the context γ and for the natural language ζ the $term_1$ may plausibly have $term_2$ occur in *role* with it (and inversely $term_2$ maintains a *co-role* relation with $term_1$). As such, they correspond closely to ORM binary fact types. Lexons are independent of specific applications and should cover relatively broad domains. Experiments are carried out to extract automatically lexons from textual material [24] and to evaluate the results [30]. Lexons are grouped by context and language. Lexons are thus to be situated on the language level.

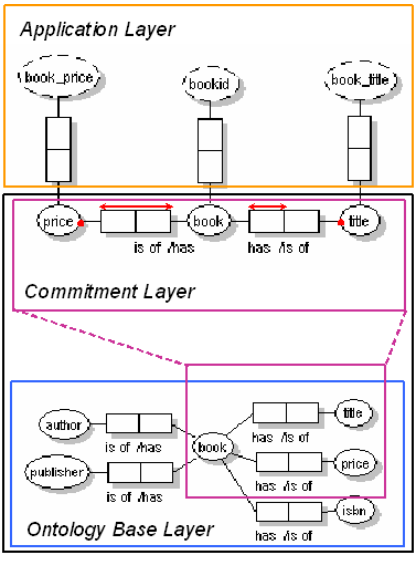


Fig. 2. The double articulation of a DOGMA ontology

Conceptual Stage

Subsequently, the *logical vocabulary is rooted in natural semantics*. The meaning of the lexon constituting parts (terms and roles) is to be determined. Existing dictionaries, thesauri, or semantic networks (e.g., (Euro)WordNet) can be used. Inevitably, new definitions will have to be created. Terminological principles and practices can be taken over [14]. Labels (short hand notation) for the notions or concepts are chosen and associated with the appropriate definition and explanation. Synonyms and translations are grouped. Mappings (depending on the language and context) are defined that link natural language words (synonyms, translations) to the corresponding concept [4]. All this has to happen in common agreement amongst the stakeholder involved, otherwise sharing of meaning will not be possible.

After which *meta-lexons* are created. These are the conceptual (i.e., language and context independent) counterparts of the lexons. Conceptual relations between concepts in a particular *domain* are represented as binary facts and constitute the ontology base (see the lower part of Figure 2). A meta-lexon can be roughly considered as two combined (inverse) RDF triples.

[4] A concept definition server that supports this functionality is being implemented in our lab.

Additional restrictions on the meta-lexons representing a specific *application* conceptualisation are situated in the commitment layer (see the middle part of Figure 2). This layer, with its formal constraints, is meant for interoperability issues between information systems, software agents and web services. These kinds of constraints are mathematically founded and concern rather typical DB schema constraints (cardinality, optionality etc.) as captured in the ORM constraints. They also correspond largely to OWL restrictions. By constraining meta-lexons instead of lexons, the "impact" of the constraints is bigger (synonyms and translations are covered). We don't present the additional steps of defining the semantic constraints here as they are straightforward for modellers familiar with ORM – see [29].

5 DOM^2 and ORM

5.1 Basic Constituents

According to Halpin, a conceptual schema of a database consists of three main constituents [12:p.31]:

- *basic fact type*s: the kinds of primitive sentences or facts
- *constraints*: the restrictions that apply to the fact types
- *derivation rules*: rules, functions or operators (including mathematical calculation or logical inference) to derive new facts from other facts.

Translated in DOGMA parlance, it means that *basic fact types* belong to the ontology base and that *constraints* belong to the commitment layer. *Derivation rules* are actually not considered as part of the actual ontology, in opposition to what other ontology researchers often claim. In the DOGMA framework, the derivation rules are situated in the application domain realm. Basically, inference rules use the logical vocabulary as it has been defined and constrained in the ontology.

5.2 Context and Language

This is an obvious point of difference between DOM^2 and ORM as ORM does not use the notion of a context and language. As explained above, the context and language constructs are needed to map natural language terms to concepts. Currently, a DOGMA context is a mere pointer to a document (or a section in a document) in which a term appears. It is valuable to have a reference to the document containing the term in its specific context of usage. Others call this pointer the co-text [14]. Another use of contexts is to group related knowledge [25:p.184]. One can expect that lexons from the same document (or parts of it) share the same background (needed for disambiguation), and very probably will be grouped in the same context in the lexon base. Algorithms, as e.g. suggested by [15], can use the "co-text" for sense disambiguation of terms. As a result, a context can be formally defined as the collection of terms (including synonyms and translations) that are associated with the concepts contained by a context [5]. More research however, is still needed on this topic, especially as quite some literature is available on contexts – e.g., [2]. An important distinction to be further developed is between the context of definition and the context of usage, which is important e.g. in an e-learning environment – see e.g. [7]). The

latter leads us to the notion of pragmatics (as in the triple syntax – semantics – pragmatics [20]). In [27], we have preliminarily suggested the use of combined sets of commitments and called these pragmatic views to capture an overall communicative situation – e.g., two intelligent agents negotiating a purchase. A context would then stand for the pragmatic situation at the ontology creation time (representing the "original" intended meaning) and is situated at the ontology base level, while the pragmatic views reflect a specific usage situation (not always foreseen and foreseeable) and are situated on the commitment level. A link with emergent semantics [1] can be made. However, due to space restrictions we leave this topic aside.

5.3 Reference Schemes

Referencing in an ORM conceptual data model happens by means of a reference scheme. E.g., a person is identified by his first name. The actual values (or strings) for the first names are stored in the database (e.g., table 'Person' with a column label 'firstname' – the object level of Figure 3). As ontologies, in principle, are not concerned with instances (=data, extension) but with meta-data (intension), referencing can only happen when an application has committed to the ontology (via lexical mapping rules) [33].

Databases that use different terms for the same notion can share data if their local database vocabulary (table and column labels) is mapped to the corresponding meaning in the ontology (being represented by a concept label). A reference scheme (linking a sense to a value type – called data type in Figure 3) now has three levels: a value type that refers to an entity type (these two belong to the conceptual schema of the information system) that is linked to a commonly defined concept label (the latter two belonging to the ontology base level), being a short hand notation for a definition of a domain notion. Value types only appear in the application layer, when legacy systems are linked to a domain ontology.

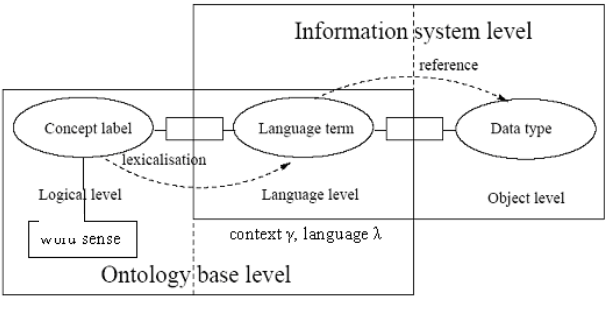


Fig. 3. Three layer reference scheme

6 Discussion

Of course, there remains a number of open questions or areas for further refinement and research. Due to the space restrictions, we only mention two pending issues. It concerns first the transformation of a lexon role and co-role into a meta-lexon relationship. Do the role and co-role have to be merged into one conceptual relationship? Or do we keep two relationships and formally consider them as separate ones? Does it make sense to keep the joint combination considering the fact that RDF triples do not

consider the inverse relationship (meaning that the co-role will always be empty when importing RDF triples into the DOGMA format) ? Currently, we choose to create two separate meta-lexons. A meta-lexon can be transformed to an RDF triple, if needed. In a later stage, the conversion of DOGMA commitments into Description Logic formulas that are used for consistency checking and implementing business logics (reasoning or inferencing) would benefit from this approach.

Another point concerns what to do with "complex concepts" (e.g., "hotel_name" vs. "airplain_manufacturing_company_name" vs. "name"). The question of naming conventions for complex concepts arises from the assumption that every concept refers to a piece of reality. Sometimes the meaning is felt to be too broad and some specialisation (expressed in natural language by a compound as 'hotel name') is wanted. Currently, we tend to reject "complex concepts", albeit it more on philosophical grounds ("notions are not to be multiplied without necessity" = Occam's razor). Practice (on a case by case basis) should show if sufficient necessity is available. This echoes the point raised by Halpin about overlapping values types.

7 Future Work and Conclusion

DOM^2, based on ORM, focuses specifically on how to model an application domain. Another, more encompassing ontology engineering life cycle, methodology called AKEM [34] is also under development at VUB STAR Lab. As both are complementary, the next aim is to integrate both into one overall ontology engineering lifecycle methodology. In order to consolidate and refine the new methodology, many modelling exercises should be undertaken in the future. We also plan to look into and refine linguistically based methods to automatically generate not only lexons [24,30] but also semantic constraints. In addition, the methodology still needs to be adapted for a collaborative modelling scenario. The ultimate goal is to provide the domain experts with a set of teachable and repeatable rules, guidelines and tools to "standardise" as much as possible an ontology engineering methodology (less art, more science).

Acknowledgment

This research was financed by the Flemish IWT 2001 #010069 "OntoBasis" project.

References

1. Aberer K., Catarci T., Cudré-Mauroux P. et al.., (2004), Emergent Semantics Systems, in, Bouzeghoub M., Goble C., Kashyap V. & Spaccapietra S.,(eds.), Proceeding of the International Conference on Semantics of a Networked World, LNCS 3226, pp. 14 – 44
2. Bouquet P., Giunchiglia F., van Harmelen F., Serafini L. & Stuckenschmidt H., (2004), Contextualizing Ontologies, Journal of Web Semantics , 26:: 1-19
3. Calvanese C., De Giacomo G., Lenzerini M., (2001), A Framework for Ontology Integration, in Proceedings of the 2001 International Semantic Web Working Symposium
4. Cunningham H., Ding Y. & Kiryakov A., (2004), Proceedings of the ISWC 2003 Workshop on Human Language Technology for the Semantic Web and Web Services

5. De Bo J. & Spyns P., Refining the notion of context within the DOGMA framework. Technical Report 12, STAR Lab, Brussel, 2003.
6. De Bo J., Spyns P. & Meersman R., (2003), Creating a "DOGMAtic" multilingual ontology infrastructure to support a semantic portal. In R. Meersman, Z. Tari et al., (eds.), On the Move to Meaningful Internet Systems 2003: OTM 2003 Workshops, LNCS 2889, pp. 253 – 266, Springer.
7. De Leenheer P. & de Moor A., Context-driven Disambiguation in Ontology Elicitation, in Shvaiko P. & Euzenat J. (eds.), (2005), Context and Ontologies: Theory, Practice and Applications: AAAI 05 Workshop, AAAI Technical Report WS-05-01, AAAI Press, pp. 17–24
8. Farrugia J., (2003), Model-Theoretic Semantics for the Web, in Proceedings of the 12th International Conference on the WWW, ACM, pp. 29 – 38
9. Genesereth M. & Nilsson N., (1987), Logical Foundations of Artificial Intelligence, Morgan Kaufmann
10. Giunchiglia F., Yatskevich M. & Giunchiglia E., (2005), Efficient Semantic Matching, in Gómez-Pérez A & Euzenat J. (eds.), The Semantic Web: Research and Applications, Proceedings of the 2nd European Semantic Web Conference, LCNS 3532, Springer, pp.272–289
11. Guarino N., (1998), Formal Ontologies and Information Systems, in Guarino N. (ed), Proc. of FOIS98, IOS Press, pp.3 – 15
12. Halpin T., (2001), Information Modeling and Relational Databases: from conceptual analysis to logical design, Morgan-Kaufmann, San Francisco.
13. Jarrar M. & Meersman R., (2002), Formal Ontology Engineering in the DOGMA Approach, in Meersman R., Tari Z. et al., (eds.), On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE; Confederated International Conferences CoopIS, DOA, and ODBASE 2002 Proceedings, LNCS 2519, Springer Verlag, pp. 1238 – 1254
14. Kerremans, K. and Temmerman, R. (2004). "Towards Multilingual, Termontological Support in Ontology Engineering". Proceedings Workshop on Terminology, Ontology and Knowledge représentation, Lyon, France, 22-23 January 2004.
15. Magnini B., Serafini L. & Speranza M., (2002), Using NLP Techniques for Meaning Negotiation, in Proceedings of the Ottavo Convegno dell'Associazione Italiana per l'Intelligenza Artificiale, (http://www-dii.ing.unisi.it/aiia2002/paper/NLP/serafini-aiia02.pdf)
16. Martinet A., (1955), Economie des changements phonétiques, Berne Francke
17. Meersman R., (1999), The Use of Lexicons and Other Computer-Linguistic Tools, in Zhang Y., Rusinkiewicz M, & Kambayashi Y., (eds.), Semantics, Design and Cooperation of Database Systems; The International Symposium on Cooperative Database Systems for Advanced Applications (CODAS 99), Heidelberg, Springer Verlag, pp. 1 – 14.
18. Meersman R., (2001), Ontologies and Databases: More than a Fleeting Resemblance, In, d'Atri A. and Missikoff M. (eds), OES/SEO 2001 Rome Workshop, Luiss Publications.
19. Meersman R., (2002), Semantic Web and Ontologies: Playtime or Business at the Last Frontier in Computing ?, In, NSF-EU Workshop on Database and Information Systems Research for Semantic Web and Enterprises, pp.61 – 67.
20. Morris Ch., (1971), Writings of the General Theory of Signs, Mouton, The Hague
21. Métais E., (2002), Enhancing information systems with natural language processing techniques, Data and Knowledge Engineering 41: 247 – 272
22. Navigli R. & Velardi P., (2004), Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites, Computational Linguistics (2): 151 – 179
23. Noy N., (2004), Semantic Integration: A Survey of Ontology-Based Approaches, SIGMOD Record Special Issue 33 [in print]

24. Reinberger M.-L. & Spyns P., (2005), Unsupervised Text Mining for the learning of DOGMA-inspired ontologies, in Buitelaar P., Cimiano Ph. & Magnini B. (eds.), Ontology Learning from Text: Methods, Applications and Evaluation, IOS Press, pp. 29 – 43
25. Sowa, J.F. (2000) Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Co
26. Spyns P., Meersman R. & Jarrar M., (2002), Data modelling versus Ontology engineering, In, Sheth A. & Meersman R. (eds.), SIGMOD Record Special Issue 31 (4): 12 – 17.
27. Spyns P. & Meersman R., From knowledge to Interaction: from the Semantic to the Pragmatic Web. Technical report 05, STAR Lab, Brussel, 2003.
28. Spyns P. & De Bo J., (2004), Ontologies: a revamped cross-disciplinary buzzword or a truly promising interdisciplinary research topic ?, Linguistica Antverpiensia NS (3): 279 – 292
29. Spyns P., (2005), Adapting the Object Role Modelling method for Ontology Modelling . In, Hacid M.-S., Murray N., Ras Z. & Tsumoto S.,(eds.), Foundations of Intelligent Systems, Proceedings of the 15th International Symposium on Methodologies for Information Systems, LNAI 3488, Springer Verlag, pp. 276 – 284
30. Spyns P. & Reinberger M.-L., (2005), Evaluating ontology triples generated automatically from texts. In, A. Gomez-Perez & Euzenat J.,(eds.), The Semantic Web: Research and Applications, Proceedings of the 2nd European Conference on the Semantic Web, LNCS 3532, Springer Verlag pp. 563 – 577
31. Stuckenschmidt H. & van Harmelen F., (2005), Information Sharing on the Web, Springer
32. van de Riet & Meersman (eds.), (1992), Linguistic Instruments in Knowledge Engineering, North Holland
33. Verheyden P., De Bo J. & Meersman R., (2004), Semantically Unlocking Database Content through Ontology-based Mediation, in Bussler C., Tannen V. & Fundulaki I. (eds.), Proceedings of the VLDB 2004 Workshops, LNCS 3372, Springer Verlag, pp. 109 – 126
34. Zhao G, Kingston J., Kerremans K., Coppens F., Verlinden R., Temmerman R. & Meersman R., (2004). Engineering an Ontology of Financial Securities Fraud, in Meersman R., Tari Z., Corrado A. et al. (eds.), Proceedings of the OTM 2004 Workshops, LNCS 3292, Springer Verlag, pp. 605 – 620

Fact Calculus: Using ORM and Lisa-D to Reason About Domains

S.J.B.A. Hoppenbrouwers, H.A. (Erik) Proper, and Th.P. van der Weide

Institute for Computing and Information Sciences, Radboud University,
Toernooiveld 1, 6525 ED Nijmegen, The Netherlands, EU
{S.Hoppenbrouwers, E.Proper, Th.P.vanderWeide}@cs.ru.nl

Abstract. We propose to use ORM and Lisa-D as means to formally reason about domains. Conceptual rule languages such as Lisa-D, RIDL and ConQuer allow for the specification of rules in a semi-natural language format that can more easily be understood by domain experts than languages such as predicate calculus, Z or OCL. If one would indeed be able to reason about properties of domains in terms of Lisa-D expressions, then this reasoning would be likely to be better accessible to people without a background in formal mathematics, such as "the average" domain expert. A potential application domain for such reasoning would be the field of *business rules*. If we can reason about business rules formulated in a semi-natural language format, the formal equivalence of (sets of) business rules (i.e. various paraphrasings) can be discussed with domain experts in a language and a fashion that is familiar to them.

1 Introduction

We will propose and explore initial ideas about a fact-based approach to reasoning based entirely on concepts that are familiar to a domain expert, as modeled through ORM/Lisa-D. We show how ORM models [5], combined with and covered by non-graphical Lisa-D expressions [6], can be subject to reasoning using an alternative type of reasoning rule. As will be explained, these reasoning rules are based on "information descriptors" [6]. However, the system is still rooted in classical predicate logic. This paper proposes to use ORM and Lisa-D as a means to formally reason about domains. Conceptual rule languages such as Lisa-D [6], RIDL [9] and ConQuer [1] allow for the specification of rules in a semi-natural language format that can be more easily understood by domain experts than languages such as predicate calculus, Z [11] or OCL [12].

A long term ambition of ours is to relate formal reasoning to styles of (communication about) reasoning close to reasoning in *contextualized* (i.e. domain-related) Natural Language (NL) [8], without loosing formal functionality. Metaphorically, fact calculus (and our communication-oriented approach to representing it) could be presented as a "next generation approach of dealing with symbol-based reasoning" as opposed to reasoning systems more directly related to formal logic. If one would indeed be able to reason about properties of domains in terms of Lisa-D expressions, then this reasoning would be likely to be

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 720–729, 2005.
© Springer-Verlag Berlin Heidelberg 2005

better accessible to people without a background in formal mathematics, such as "the average" domain expert. A potential application domain for such reasoning would be the field of *business rules*. When reasoning can be done at the level of business rules formulated in a semi-natural language format, the equivalence of (sets of) business rules can be discussed with domain experts in a language that is familiar to them.

The aim of the modeling process as we see it [8] is to find a representation mechanism for sentences from some domain language. The result may be seen as a signature for a formal structure. Each elementary fact type is a relation in this structure. Assuming a set of variables, we can introduce the expressions over this signature [2]. This signature forms the base of a formal theory about a domain. The constraints then are seen as the axioms of this theory.

In classic reasoning (for example, natural deduction [3]), formal variables are used, through which we can formulate statements and try to prove them from the axioms, using a conventional reasoning mechanism (containing for example *modus ponens*). In this paper we explore an alternative approach, based on specific models (populations), and focus on properties of some particular population.

Statements about a current population can be represented as Lisa-D statements. For clarity's sake, we include "translations" in regular English for every Lisa-D statement. The translations sometimes leave out redundant information that is nevertheless vital in reasoning about information descriptors. Though such translation cannot currently be produced deterministically or even automatically, we do intend to explore ways of achieving this in the future.

For example, consider the following statement, that could be a query:

EACH Student living in City 'Elst' MUST ALSO BE attending Course 'Modeling'
Each student that lives in the city of Elst also attends the course Modeling

2 Fact Calculus

2.1 Starting from Predicate Calculus

In existing approaches, reasoning in terms of a conceptual model is closely related to reasoning in predicate calculus [2, 6]. This form of reasoning is instance related, for example transitivity of the relation f is expressed as:

$$\forall_{x,y,x} [f(x,y) \wedge f(y,z) \Rightarrow f(x,z)]$$

Using instances leads to a style of reasoning that may be qualified as *reasoning* **within** *the Universe of Discourse*. Using a conceptual language such as Lisa-D provides the opportunity to reason without addressing particular instances. This may be characterized as *reasoning* **about** *the Universe of Discourse*.

First it should be noted that a precise formulation of domain rules may require a way of referring to general instances in order to describe concisely their relation. In daily practice people use NL mechanisms to make such references. However, in many situations domain rules can be nicely formulated without addressing any particular instance. As an example, consider the rule (phrased in NL):

When a car is returned, then its official documents should also be returned

In this sentence, without explicitly addressing any particular car, the subtle use of the reference *its* provides a sufficient reference.

The main idea behind Lisa-D, as present in its early variant RIDL [9] is a functional, variable-less description of domain-specific information needs. In Lisa-D the mechanism of variables is of a linguistic nature. Variables are special names that can be substituted once they are evaluated in a context that generates values for this variable. The set comprehension construct is defined in that way. Lisa-D expressions are based on a domain-specific lexicon, that contains names for the elements that constitute a conceptual schema. The lexicon contains a name for each object type, and also provides names for the construction mechanism in a conceptual schema (such as the roles and object types it contains).

2.2 Information Descriptors

The names in the lexicon are on par with the words from which NL sentences are constructed. Lisa-D sentences are referred to as information descriptors. The base construction for sentences is juxtaposition. By simply concatenating information descriptors, new information descriptors are constructed. Before describing the meaning of information descriptors, we will first discuss how such descriptors will be interpreted.

The semantics of Lisa-D can be described in various ways. The simplest form is its interpretation in terms of set theory. Other variants use bags, fuzzy sets, or probabilistic distributions. In order to make a bridge with predicate calculus, we will view information descriptors as binary predicates.

Let D be an information descriptor, and P a population of the corresponding conceptual schema. We then see this information descriptor as a binary predicate. Let $\mathcal{V}$ be a set of variables, then we write $P \models x \; [\![D]\!] \; y$ to express that in population P there is a relation between x and y via D, where $x, y \in \mathcal{V}$. For each object type O we introduce a unary predicate: $P \models O(x)$ iff $x \in P(O)$ and for each role R of some fact type F we introduce a binary predicate: $P \models R(x, y)$ iff $y \in P(F) \wedge x = y(R)$. Note that the instances of a fact type (y) are formally treated as functions from the roles of fact type F to instances of the object types involved in a role. In other words, $y(R)$ yields the object playing role R in fact y.

At this point we can describe the meaning of elementary information descriptors as follows. Let o be the name of object type O and r the name of a role R, then o and r are information descriptors with semantics:

$$x \,[\![o]\!]\, y \;\triangleq\; O(x) \wedge x = y \qquad\qquad x \,[\![r]\!]\, y \;\triangleq\; R(x,y)$$

Roles involved in fact types may actually receive multiple names. This is illustrated in figure 1. A single role may, in addition to its 'normal' name, also receive a *reverse role name*. Let v be the reverse role name of role R, then we have:

$$x \,[\![v]\!]\, y \triangleq R(y,x)$$

Any ordered pair of roles involved in a fact type may receive a *connector name*. The connector names allow us to 'traverse' a fact type from one of the participating object types to another one. If c is the connector name for a role pair $\langle R, S\rangle$, then the semantics of the information descriptor c are defined as:

$$x \,[\![c]\!]\, z \triangleq \exists_y \,[R(x,y) \wedge S(y,z)]$$

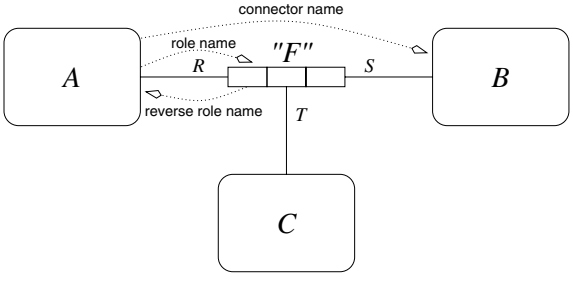


Fig. 1. Role names

Elementary information descriptors can be composed into complex information descriptors using constructions such as *concatenation*, *conjunction*, *implication*, *disjunction* and *complement*. These may refer to the fronts alone or both fronts and tails of descriptors. In this paper we will use:

$$\begin{aligned}
x \,[\![D_1\ D_2]\!]\, y &\triangleq \exists_z \,[x \,[\![D_1]\!]\, z \wedge z \,[\![D_2]\!]\, y] \\
x \,[\![D_1 \,\mathsf{AND\ ALSO}\, D_2]\!]\, y &\triangleq \exists_z \,[x \,[\![D_1]\!]\, z] \wedge \exists_z \,[x \,[\![D_2]\!]\, z] \wedge x = y \\
x \,[\![D_1 \,\mathsf{MUST\ ALSO\ BE}\, D_2]\!]\, y &\triangleq \exists_z \,[x \,[\![D_1]\!]\, z] \Rightarrow \exists_z \,[x \,[\![D_2]\!]\, z] \wedge x = y \\
x \,[\![D_1 \,\mathsf{OR\ IS}\, D_2]\!]\, y &\triangleq \exists_z \,[x \,[\![D_1]\!]\, z] \vee \exists_z \,[x \,[\![D_2]\!]\, z] \wedge x = y \\
x \,[\![D_1 \,\mathsf{BUT\ NOT}\, D_2]\!]\, y &\triangleq \exists_z \,[x \,[\![D_1]\!]\, z] \wedge \neg \exists_z \,[x \,[\![D_2]\!]\, z] \wedge x = y
\end{aligned}$$

where D_1 and D_2 are information descriptors and x, y and z are variables. Some example expression would be:

Person working for Department 'I&KS'
People working for department 'I&KS'

Person (working for Department 'I&KS' AND ALSO owning Car of Brand 'Seat')
People working for department 'I&KS' who also own a car of brand Seat

Person (working for Department 'I&KS' MUST ALSO BE owning Car of Brand 'Seat')
People who, if they work for department 'I&KS', also own a car of brand 'Seat'

Person (owning Car of Brand 'Seat' OR IS living in City 'Nijmegen')
People who own a car of brand Seat, or live in the city of Nijmegen

Person (working for Department 'I&KS' BUT NOT living in City 'Amsterdam')
People working for department 'I&KS', but who do not live in the city of Amsterdam

Correlation operators form a special class of constructs:

$$x \,[\![D \,\mathsf{THAT}\, o]\!]\, y \triangleq x \,[\![D \; o]\!]\, y \wedge x = y$$
$$x \,[\![D \,\mathsf{MUST\ BE\ ANOTHER}\, o]\!]\, y \triangleq x \,[\![D \; o]\!]\, y \wedge x \neq y$$

where D is an information descriptor and o is the name of an object type. Some examples of its use are:

Person working for Department having as manager THAT Person
People who work for a department that has that person as a manager

Person owning Car having Brand being of Car being owned by MUST BE ANOTHER Person
People who own a car of the same brand as another person's car

To make some Lisa-D expressions more readable, we also introduce dummy words AN and A which have no real meaning:

$$\mathsf{AN}\, P \;\triangleq\; \mathsf{A}\, P \;\triangleq\; P$$

Using these 'dummy words' the last two Lisa-D expressions can be re-phrased as:

Person working for A Department having as manager THAT Person
Person owning A Car having Brand being of A Car being owned by MUST BE ANOTHER Person

In Lisa-D many more constructions exist to create complex information descriptors. However, in this paper we limit ourselves to those constructions that are needed for the considerations discussed below. Note again that the natural language likeness of the Lisa-D expressions used in this paper can be improved considerably. For reasons of compactness, this paper defined fact calculus directly in terms of (verbalizations of) information descriptors. However, in the original Lisa-D path expressions were used as the underlying skeleton for rules, where the information descriptors 'merely' serve as the flesh on the bones. Using linguistic techniques as described in e.g. [7, 4] this 'flesh' can obtain a more natural structure. Future work will also concentrate on improved verbalizations of path expressions.

2.3 Rules

Lisa-D has a special way of using information descriptors to describe rules that should apply in a domain. These rules can be used to express constraints and/or business rules. We will use the more general term *rule* for such expressions. These rules consist of information descriptors that are interpreted in a boolean way;

i.e. if no tuple satisfies the predicate, the result is false, otherwise it is true. This leads to the following semantics for rules:

$$\begin{aligned}
[\![\mathsf{EACH}\ D]\!] &\triangleq \forall_x \exists_y [x\ [\![D]\!]\ y] & [\![\mathsf{SOME}\ D]\!] &\triangleq \exists_{x,y} [x\ [\![D]\!]\ y] \\
[\![R_1\ \mathsf{AND}\ R_2]\!] &\triangleq [\![R_1]\!] \wedge [\![R_2]\!] & [\![R_1\ \mathsf{OR}\ R_2]\!] &\triangleq [\![R_1]\!] \vee [\![R_2]\!] \\
[\![R_1\ \mathsf{IMPLIES}\ R_2]\!] &\triangleq [\![(\mathsf{NOT}\ R_1)\ \mathsf{OR}\ R_2]\!] & [\![\mathsf{NOT}\ R_1]\!] &\triangleq \neg [\![R_1]\!] \\
[\![\mathsf{NO}\ D]\!] &\triangleq [\![\mathsf{NOT\ SOME}\ D]\!]
\end{aligned}$$

where D is an information descriptor and R_1, R_2 are rules.

Note that the $\exists$ and $\forall$ quantifications in the EACH D and SOME D constructs range over all possible instances. Limiting a variable to a specific class of instances is done similar to set theory, where:

$$\forall_{x \in D} [P(x)] \triangleq \forall_x [x \in D \Rightarrow P(x)] \triangleq \forall_x [D(x) \Rightarrow P(x)]$$

In our case we would typically write: EACH T MUST ALSO BE C where T is an information descriptor representing the domain over which one ranges and C is the condition.

In the context of rules, we will also use the following syntactic variations of THAT and MUST BE ANOTHER:

$$\begin{aligned}
x\ [\![D\ \mathsf{THAT}\ o]\!]\ y &\triangleq x\ [\![D\ \mathsf{MUST\ BE\ THAT}\ o]\!]\ y \\
x\ [\![D\ \mathsf{MUST\ BE\ ANOTHER}\ o]\!]\ y &\triangleq x\ [\![D\ \mathsf{ANOTHER}\ o]\!]\ y
\end{aligned}$$

3 Examples of Rule Modeling

In this section we provide some illustrations of ways one can reason within the fact calculus. At the moment most of the reasoning can be done by 'jumping down' to the level of predicate calculus. It indeed makes sense to also introduce derivation rules at the information descriptor and rules level. This is, however, beyond the scope of the short discussion provided in this paper.

3.1 Example: Trains and Carriages

As a first example of the use of fact calculus to reason about domains, consider the following two rules:

EACH Train MUST ALSO BE consisting of A Carriage
Each train consists of carriages

EACH Carriage MUST ALSO BE having A Class
Each carriage has a class.

Using predicate calculus based inference, one could infer:

EACH Train MUST ALSO BE consisting of A Carriage having A Class
Each train consists of carriages that have a class.

3.2 Example: Return of Cars and Papers

Now consider the rule (phrased in NL) that was mentioned earlier in this paper.

When a car is returned, then its official documents should also be returned.

First we note that this rule may be characterized as an action rule; it *imperatively* describes what actions are required when returning a car. From our perspective, however, we prefer to focus on *declarative* characterizations of the underlying Universe of Discourse, as also advocated in [10–article 4].

As a first step, we formulate a declarative characterization. In some cases a positive formulation works best; in this case we choose to use negation, and derive from the action rule the situation that the follow up action of this rule is intending to avoid:

NO Car being returned AND ALSO having (AN Official-paper BUT NOT being returned)
In no case a car may be returned, if papers belonging to that car are not also returned.

In order to make this expression more readable it is sensible to re-balance the order of appearance of the types in the expression. By focussing on the role of the Official-papers we would get:

NO Official-paper of A Car being returned BUT NOT being returned
In no case may papers of a returned car not also be returned.

Using predicate calculus we can formally declare the equivalence of these statements, while the latter formulation (paraphrasing) is more natural.

Via a negative to positive transformation we can also obtain:

EACH Official-paper from A Car being returned MUST ALSO BE being returned
All papers from returned cars must also be returned.

3.3 Example: Car Registration

As another example, we consider the reasoning that leads to the conclusion that: each person has a license plate, under the assumption that each person has a car and that each car has a license plate. We are thus given the following rules:

1. Each person must own a car
2. Each car must be registered by a licence plate

The resulting rules will serve as *domain axioms* for our reasoning system:

EACH Person MUST ALSO BE owning Car
Each person must own a car

EACH Car MUST ALSO BE being registered by License plate
Each car must be registered by a licence plate

The reasoning rule we apply to combine these two rules is the *First Implication Rule*:

$$\frac{D_1 \text{ MUST ALSO BE } D_2\ D_3 \qquad D_3 \text{ MUST ALSO BE } D_4}{D_1 \text{ MUST ALSO BE } D_2\ (D_3 \text{ MUST ALSO BE } D_4)}$$

where D_1, D_2, D_3 and D_4 are information descriptors. Before proceeding with the example, we first will show the validity of the First Implication Rule (see figure 2). Suppose $x \, [\![D_1]\!] \, p$ for some x and p. Then we can conclude from the first rule that also $x \, [\![D_2 \; D_3]\!] \, r$ for some r, and thus for some q we have $q \, [\![D_3]\!] \, r$. Applying the second rule leads to the conclusion $q \, [\![D_4]\!] \, y$ for some y. Combining the arguments leads to: $x \, [\![D_2 \; (D_3 \, \mathsf{MUST\ ALSO\ BE}\, D_4)]\!] \, y$.

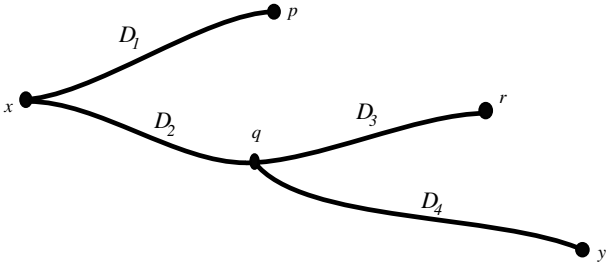


Fig. 2. Situation

Applying the above two reasoning rules by setting:

$D_1 =$ Person $\quad D_2 =$ owning
$D_3 =$ Car $\quad D_4 =$ being registered by License plate

we obtain the validity of the following expression in each population:

EACH Person MUST ALSO BE owning (Car MUST ALSO BE being registered by License plate)
Each person must own a car, and that car must be registered by a licence plate.

Furthermore, we also have the following *Absorbtion Rule*:

$$\frac{o \, \mathsf{MUST\ ALSO\ BE}\, D}{o \; D}$$

where D is an information descriptor, while o is an object type name. This rule allows us to rewrite the previous expression to:

EACH Person MUST ALSO BE owning Car being registered by License plate
Each person must own a car that must be registred by a licence plate.

3.4 Example: Car Ownership

Next we focus on uniqueness within binary relationships. The following rule expresses the rule that a person may not own more than one car:

NO Car being owned by A Person owning MUST BE ANOTHER Car

No car may be owned by a person who also owns another car

With respect to binding rules, the concatenation operator has the highest priority. Furthermore, the MUST BE ANOTHER operator has a higher priority than the NO operator. As a consequence, we get the following binding structure:

NO ((Car being owned by A Person owning) MUST BE ANOTHER Car)

We first will transform this rule into a positive formulation. This positive formulation is better suited for reasoning, but is less obvious to understand from an intuitive point of view.

EACH Car being owned by Person owning Car MUST BE THAT Car
Each car that is owned by a person owning a car, must be this person's only car

Next we add the rule that a licence plate may be associated with at most one car, or in its positive formulation:

EACH Licence plate registering A Car being registered by A Licence plate MUST BE THAT License plate
Each licence plate that registers a car, must be a unique licence plate for that car

In order to combine these two uniqueness constraints, we apply the following reasoning rule:

$$\frac{\text{EACH } D_1\ o_1\ D_2 \text{ MUST BE THAT } o_2 \qquad \text{EACH } D_3 \text{ MUST BE THAT } o_1}{\text{EACH } D_1\ D_3\ D_2 \text{ MUST BE THAT } o_2}$$

where D_1, D_2 and D_3 are information descriptors, while o_1 and o_2 are names of object types. We first prove the validity of this rule. Presume the rules EACH $D_1\ o_1\ D_2$ MUST BE THAT o_2 and EACH D_3 MUST BE THAT o_1, and assume $x\ [\![D_1\ D_3\ D_2]\!]\ y$, then from the definition of concatenation we know that for some p and q we have $x\ [\![D_1]\!]\ p\ [\![D_3]\!]\ q\ [\![D_2]\!]\ y$. From $p\ [\![D_3]\!]\ q$ we conclude that also $p\ [\![o_1]\!]\ q$, and thus $x\ [\![D_1\ o_2\ D_2]\!]\ y$. Applying the first rule yields $x\ [\![o_2]\!]\ y$. Applying this reasoning rule by setting:

D_1 = License plate registering — o_1 = Car
D_2 = being registered by License plate — o_2 = Licenseplate
D_3 = Car being owned by Person owning Car

we get:

EACH Licence plate registering A Car being owned by A Person owning A Car being registered by A Licence plate MUST BE THAT License plate
Each licence plate that registers a car (owned by a person), must be a unique licence plate for that car owned by that person

As a negative formulated sentence:

NO Licence plate is registering A Car being owned by A Person owning A Car being registered by ANOTHER Licence plate
No licence plate may register a car (that is owned by a person), that is also registered by another licence plate

4 Discussion and Conclusion

In this paper we proposed to use ORM and Lisa-D as a means to formally reason about domains. We explored some aspects of such reasoning. During conceptual modeling, a modeler can add consistency rules that describe the intended populations of the conceptual schema. We assumed that such rules can

be formulated in the conceptual language Lisa-D. We discussed reasoning with Lisa-D expressions. Especially, we speculated that such reasoning may well be closer to the way people naturally reason about specific application domains than more traditional forms of (formal) reasoning. If this is indeed the case, we may be able to "reverse the modeling process", and focus on sample reasoning in the application domain, deriving from explicit reasoning examples an underlying system of reasoning rules and domain-specific axioms.

In the near future the original definition of Lisa-D will be adapted to better suit the needs for formal reasoning about domains. More work is also needed in providing more natural verbalisation/paraphrasing of Lisa-D expressions, more specifically the verbalisation of *path expressions* as mentioned in section 2.2. A link to formal theorem proving tools will be considered as well.

References

1. A.C. Bloesch and T.A. Halpin. ConQuer: A Conceptual Query Language. In B. Thalheim, ed., *Proc. of the 15th Intl Conference on Conceptual Modeling (ER'96)*, volume 1157 of *LNCS*, pages 121–133, Cottbus, Germany, EU, 1996.
2. P. van Bommel, A.H.M. ter Hofstede, and Th.P. van der Weide. Semantics and verification of object-role models. *Information Systems*, 16(5):471–495, 1991.
3. H. B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, New York, New York, USA, 1972.
4. P.J.M. Frederiks. *Object-Oriented Modeling based on Information Grammars*. PhD thesis, University of Nijmegen, Nijmegen, The Netherlands, EU, 1997.
5. T.A. Halpin. *Information Modeling and Relational Databases, From Conceptual Analysis to Logical Design*. Morgan Kaufman, San Mateo, California, USA, 2001.
6. A.H.M. ter Hofstede, H.A. (Erik) Proper, and Th.P. van der Weide. Formal definition of a conceptual language for the description and manipulation of information models. *Information Systems*, 18(7):489–523, October 1993.
7. J.J.A.C. Hoppenbrouwers. *Conceptual Modeling and the Lexicon*. PhD thesis, Tilburg University, Tilburg, The Netherlands, EU, 1997. ISBN 90-5668-027-7
8. S.J.B.A. Hoppenbrouwers, H.A. (Erik) Proper, and Th.P. van der Weide. Fundamental understanding of the act of modelling. Proceedings of the 24th International Conference on Conceptual Modeling (ER2005), Klagenfurt, Austria, EU.
9. R. Meersman. The RIDL Conceptual Language. International Centre for Information Analysis Services, Control Data Belgium, Inc., Brussels, Belgium, 1982.
10. R.G. Ross, editor. *Business Rules Manifesto*. Business Rules Group, November 2003. Version 2.0. `http://www.businessrulesgroup.org/brmanifesto.htm`
11. J.M. Spivey. *Understanding Z: A Specification Language and its Formal Semantics*. Cambridge University Press, Cambridge, United Kingdom, EU, 1988.
12. J. Warmer and A. Kleppe. *The Object Constraint Language: Getting Your Models Ready for MDA*. Addison-Wesley, Reading, Massachusetts, USA, 2nd edition, 2003.

Schema Equivalence as a Counting Problem

H.A. (Erik) Proper and Th.P. van der Weide

Institute for Computing and Information Sciences,
Radboud University Nijmegen,
Toernooiveld 1, 6525 ED Nijmegen, The Netherlands, EU
{E.Proper, Th.P.vanderWeide}@cs.ru.nl

Abstract. In this paper we introduce some terminology for comparing the expressiveness of conceptual data modeling techniques, such as ER, NIAM, PSM and ORM, that are finitely bounded by their underlying domains. Next we consider schema equivalence and discuss the effects of the sizes of the underlying domains. This leads to the introduction of the concept of finite equivalence, which may serve as a means to a better understanding of the fundamentals of modeling concepts (utility). We give some examples of finite equivalence and inequivalence in the context of ORM.

1 Schema Equivalence

When modeling a Universe of Discourse ([ISO87]), it is generally assumed that we can recognize *stable states* in this Universe of Discourse, and that there are a number of actions that result in a change of state (*state transitions*). This is called the state-transition model. Furthermore we assume that the Universe of Discourse has a unique *starting state*.

In mathematical terms, a Universe of Discourse U∘D consists of a set $\mathcal{S}$ of states, a binary transition relation τ over states, and an initial state $s_0 \in \mathcal{S}$:

$$\text{U}\circ\text{D} = \langle \mathcal{S}, \tau, s_0 \rangle$$

The purpose of the modeling process is to construct a formal description, (a *specification*) Σ of U∘D, in terms of some underlying formalism. This specification will have a component $\mathcal{S}(\Sigma)$ that specifies $\mathcal{S}$, a component $\tau(\Sigma)$ that specifies τ, and a state $s_0(\Sigma)$ that is designated as the initial state s_0.

The main requirement for specification Σ is that it behaves like U∘D. This can be shown by a (partial) function h, relating the states from $\mathcal{S}(\Sigma)$ to the (real) states $\mathcal{S}$ from U∘D such that h shows this similarity. Such a function is called a *(partial) homomorphism*. If each state of U∘D is captured by the function h, we call Σ a *correct specification* with respect to U∘D, as each state of U∘D has a representation in Σ. In that case, the function h is surjective, and called an *epimorphism* (see also [Bor78]).

Definition 1. *We call h a* partial *homomorphism between Σ and U∘D if*

1. *h is a (partial) function $h : \mathcal{S}(\Sigma) \rightarrow \mathcal{S}$*

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 730–739, 2005.
© Springer-Verlag Berlin Heidelberg 2005

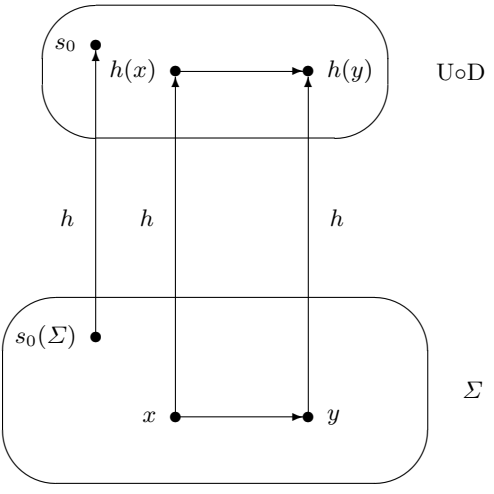


Fig. 1. A correct specification

2. transitions commute under h:

$$\forall_{s,t \in \mathcal{S}(\Sigma)} \left[\langle s,t \rangle \in \tau(\Sigma) \Leftrightarrow \langle h(s), h(t) \rangle \in \tau \right]$$

3. h maps the initial state of the specification onto the initial state of U∘D:

$$h(s_0(\Sigma)) = s_0$$

If h is surjective, we call h an epimorphism between Σ and U∘D.

We call an algebra $\mathcal{A}$ (partially) homomorphic with algebra $\mathcal{B}$, if there exists a (partial) homomorphism from $\mathcal{A}$ into $\mathcal{B}$. If schema Σ is a description of $\mathcal{A}$, then we will also call Σ (partially) homomorphic with $\mathcal{B}$. The notion of epimorphism is extended analogously.

Note that in a correct specification Σ, a state of U∘D may have more than one corresponding state in $\mathcal{S}(\Sigma)$. In that case we have a redundant representation for the states of U∘D. Redundant representations are useful as they provide opportunities for improvement of efficiency.

The disadvantage of a redundant representation is that we do not have a description of U∘D that is free of implementation (representation) details. A description can only be implementation independent if each state has a unique representant. Such a description is called a *conceptual schema* in the context of information systems. This is the case if the function h that relates Σ to U∘D is bijective.

The *expressiveness* of a formal method $\mathcal{M}$ is introduced as the set of "U∘D"'s it can model. This can be described by:

$$\left\{ \langle \mathcal{S}(\Sigma), \tau(\Sigma), s_0(\Sigma) \rangle \;\middle|\; \Sigma \in \mathcal{L}(\mathcal{M}) \right\}$$

If we restrict ourselves in this definition to $\tau(\Sigma) = \emptyset$, we get the so-called *base expressiveness* of method $\mathcal{M}$. The base expressiveness usually is the criterion that is used intuitively when comparing different methods.

From the above definition of conceptual schema, the following definition of schema equivalence can be derived.

Definition 2. *Two specifications Σ and Σ' are equivalent, $\Sigma \cong \Sigma'$, if there exists a homomorphism h from Σ onto Σ' such that h is a bijection.*

2 Schema Equivalence in ORM

In this section we consider the base expressiveness of ORM. We focus on the formal definition as for example specified in the Predicator Model (PM, see [BHW91]), and discuss schema equivalence in that context. Within that context, we may omit differences between ORM and PM.

Let Σ be an ORM schema, with underlying label type set $\mathcal{L}$, then this schema specifies the following set of states:

$$\mathcal{S}(\Sigma) = \{p \mid \mathsf{IsPop}_{\mathcal{L}}(\Sigma, p)\}$$

A population p is a function assigning a set of instances to each object type in schema Σ. The $\mathsf{IsPop}_{\mathcal{L}}$ predicate determines whether p is a proper population. The population of label values is restricted to values of some domain $\mathcal{D}$. We will show that the base expressiveness strongly depends on the actual choice of $\mathcal{D}$. In this restricted sense the resulting state space of schema Σ is:

$$\mathcal{S}_{\mathcal{D}}(\Sigma) = \{p \mid \mathsf{IsPop}_{\mathcal{L}}(\Sigma, p) \wedge \forall_{x \in \mathcal{L}} [p(x) \subseteq \mathcal{D}]\}$$

Please note that we restrict ourselves to finite populations, i.e., populations where each object type x has assigned a finite population ($p(x) < \infty$). Using this definition we introduce the notion of domain equivalence.

Definition 3. *Two ORM schemas Σ and Σ' are domain equivalent over domain $\mathcal{D}$, denoted as $\Sigma \cong_{\mathcal{D}} \Sigma'$, if they have equivalent state spaces:*

$$\mathcal{S}_{\mathcal{D}}(\Sigma) \cong \mathcal{S}_{\mathcal{D}}(\Sigma')$$

The sets A and B are called equivalent ($A \cong B$) if there exists a bijection from A into B.

Note that being equivalent does not mean that both schemata are as suitable in representing U∘D. For suitability we should take the complexity of the transition into account. This complexity however is outside the scope of this paper.

A first result is that the underlying domain may be such expressive that any two schemata based on that domain are equivalent:

Lemma 1. *Let Σ and Σ' be ORM schemas then:*

$$\mathcal{D} \text{ countably infinite } \Rightarrow \Sigma \cong_{\mathcal{D}} \Sigma'$$

Proof: We will only give a brief outline of this proof. The important step is to prove that the number of populations in a schema with a countable domain is countable itself (assuming finite populations). This however, is true because every population can be coded as a finite string by ordering the object types in the schema at hand and listing their populations sequentially, according to this ordering, separated by special separator symbols. Each such finite string can uniquely be translated to a finite bitstring, which can be considered as a natural number in binary representation. □

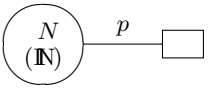


Fig. 2. The most simple universal schema

We conclude that the expressiveness of ORM data structuring in the context of a countably infinite domain is limited as all schemata are equivalent in that case. Note that, in the context of countably infinite domains, this property holds for most other data models as well. Each schema thus can be considered as a universal schema, as it is expressive enough to "simulate" any other schema. The analogon of a universal schema in the algorithmic world is the universal Turing machine (see for example [CAB+72]). The most simple universal schema is shown in figure 2. This simple schema can represent any population of any other schema, by using the counting schema described in the proof above. The role of the unary fact type is to exclude all elements from N that do not correspond to a valid population of the simulated schema.

We restrict ourselves to a finite domain for label values. As a direct consequence, schema Σ has a finite state space. We introduce the notion of finite equivalence:

Definition 4. *Two ORM schemata Σ and Σ' are finite equivalent denoted as $\Sigma \cong_f \Sigma'$, if they have an equivalent state space for equivalent finite underlying domains, or if for all $\mathcal{D}$ and $\mathcal{D}$':*

$$\mathcal{D} \cong \mathcal{D}' \wedge |\mathcal{D}| < \infty \Rightarrow \mathcal{S}_{\mathcal{D}}(\Sigma) \cong \mathcal{S}_{\mathcal{D}'}(\Sigma')$$

Finite equivalence can be proven by the construction of a bijection between the two state spaces of the schemas.

Example 1. The schemas Σ and Σ' from figure 3, are finite equivalent.

Proof: *The basic idea is to define a translation from instances from Σ to instances from Σ' such that we have a bijection between $\mathcal{S}(\Sigma)$ and $\mathcal{S}(\Sigma')$. This is achieved by relating identical instances of object types A, B and C in both schemas and instances $\{p : a, q : b\}$ in $\mathsf{Pop}(f)$ and $\{r : \{p : a, q : b\}, s : c\}$ in $\mathsf{Pop}(g)$ to one instance $\{t : a, u : b, v : c\}$ in $\mathsf{Pop}(h)$.*

Note the importance of the total role (the black dot) on predicator r in this transformation. Its semantics is:

$$x \in \mathsf{Pop}(f) \Rightarrow \exists_{y \in \mathsf{Pop}(g)} [y(r) = x]$$

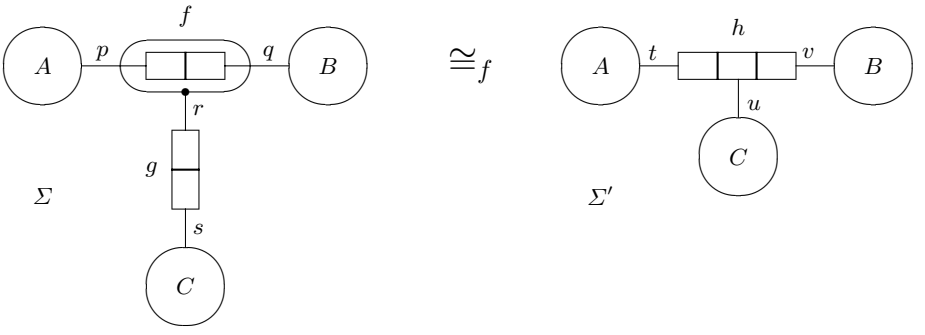


Fig. 3. Example of finite equivalence

Therefore, the total role makes it unnecessary to consider instances of fact type f that do not contribute in fact type g. For a general definition of the semantics of constraints in NIAM schemas, refer to [BHW91]. □

Finite inequivalence can be proven by showing that the state spaces of the underlying schemas are not equal in size.

Example 2. If we omit the total role from schema Σ in figure 3, the schemas are not *finite equivalent.*

Proof: *Let a, b and c be the population size of A, B and C respectively. The number of populations of fact type f amounts to:*

$$\sum_{i=0}^{ab} \binom{ab}{i} = 2^{ab}$$

Now suppose f is populated with i tuples, then for g we can have 2^{ic} *different populations. The number of populations of Σ therefore amounts to:*

$$\begin{aligned} \sum_{i=0}^{ab} \binom{ab}{i} 2^{ic} &= \sum_{i=0}^{n} \binom{ab}{i} (2^c)^i \\ &= (1+2^c)^{ab} \end{aligned}$$

On the other hand, the number of populations of Σ′ equals $2^{abc} = (2^c)^{ab}$. □

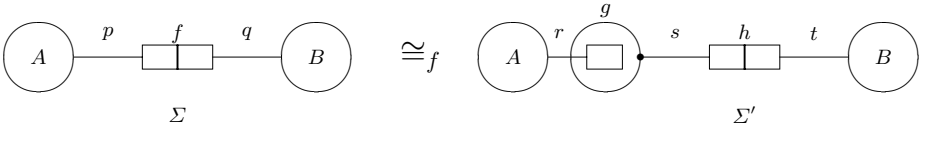


Fig. 4. Another example of finite equivalence

Example 3. In figure 4, another example of finite equivalence is shown.

Proof: *The main observation is that instances occurring in predicator p of schema Σ are to be mapped onto identical instances in the population of fact type g in schema Σ'. Instances of object types A and B in both schemas are again related via an identical mapping. Instances in fact type f in schema Σ are related to identical instances in fact type h in schema Σ'.* □

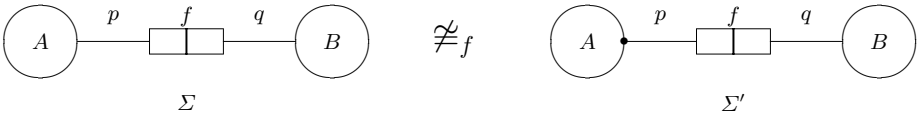


Fig. 5. Example of finite inequivalence

Example 4. In figure 5 two schemas are depicted, which are not finite equivalent.

Proof: *It is not hard to see that the number of populations in Σ with $|\mathsf{Pop}(A)| = a$ and $|\mathsf{Pop}(B)| = b$ is $(2^b)^a$, while the number of populations in Σ' with the same restriction is $(2^b - 1)^a$.* □

3 An Upper Bound for Populatability

A data modeling technique is called finitely bounded by its underlying domains, if each schema from that technique allows for a finite number of populations, in case of a finite domain of label values.

Definition 5. *The populatability of a schema Σ is:*

$$m_{\mathcal{D}}(\Sigma) = \left\| \mathcal{S}_{\mathcal{D}}(\Sigma) \right\|$$

As each schema can be populated by the empty population ([BHW91]), an immediate consequence is:

Lemma 2.

$$\|\mathcal{D}\| = 0 \Rightarrow \forall_{\Sigma \in \mathcal{L}(\mathcal{M})} \left[m_{\mathcal{D}}(\Sigma) = 1 \right]$$

Definition 6. *Method $\mathcal{M}$ is called finitely bounded by its underlying domains $\mathcal{D}$ if:*

$$\|\mathcal{D}\| < \infty \Rightarrow \forall_{\Sigma \in \mathcal{L}(\mathcal{M})} \left[m_{\mathcal{D}}(\Sigma) < \infty \right]$$

In this section we derive an upper bound on the populatability of a schema. In order to simplify the derivation, we restrict ourselves to fact schemata, i.e., schemata Σ without entity types (i.e., $\mathcal{E}(\Sigma) = \emptyset$).

Lemma 3.

$$\forall_{\Sigma} \exists_{\Sigma'} \left[\Sigma \equiv \Sigma' \wedge \mathcal{E}(\Sigma') = \emptyset \right]$$

Proof: Replace each entity type by a fact type, corresponding to its identification. If the identification of entity type x consists of the convolution of k path expressions (i.e., $mult(x) = k$, see [HPW93]), then this replacement leads to the introduction of a k-ary fact type. The resulting schema is denoted as $de(\Sigma)$. Then obviously $\Sigma \equiv de(\Sigma)$ and $\mathcal{E}(de(\Sigma)) = \emptyset$. □

The number $p(de(\Sigma))$ of predicators of schema $de(\Sigma)$ is found by:

Lemma 4.

$$p(de(\Sigma)) = p(\Sigma) + \sum_{x \in \mathcal{E}(\Sigma)} mult(x)$$

Proof: Obvious!

Next we introduce a series $\{N_p\}_{p \geq 0}$ of schemata (see figure 6), consisting of a single p-ary fact type over some label type L. These schemata are the best populatable schemata among schemata with the same number of predicators. □

Theorem 1.

$$\|\mathcal{D}\| > 1 \Rightarrow \forall_{\Sigma} \left[m(\Sigma) \leq m(N_{p(de(\Sigma))}) \right]$$

Proof: First we remark $m(\Sigma) = m(de(\Sigma))$. Next we use the fact that a schema becomes better populatable by undeeper nesting of (at least) binary fact types. This is shown in lemma 5. Furthermore, merging fact types improves populatability (see lemma 7). By repeatedly applying these steps, schema $N_{p(de(\Sigma))}$ will result. □

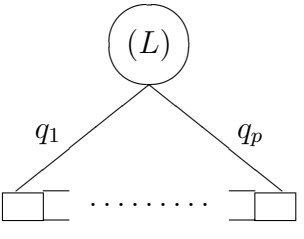


Fig. 6. Best populatable schemata

Lemma 5. *Consider the schemata Σ_1, Σ_2 and Σ_3 from figure 7, then:*

$$\|\mathcal{D}\| > 1 \Rightarrow m(\Sigma_1) \leq m(\Sigma_2) \leq m(\Sigma_3)$$

Table 1. Growth of populatability

n	$m(\Sigma_1)$	$m(\Sigma_2)$	$m(\Sigma_1)$
0	1	1	1
1	3	3	2
2	21	81	256
3	567	19683	134217728
4	67689	43046721	1.845E+19

Proof: Let $\|\mathcal{D}\| = n$, then:

$$\begin{aligned} m(\Sigma_1) &= \sum_{i=0}^{n} \binom{n}{i} \sum_{j=0}^{i} \binom{i}{j} 2^{(j^2)} \\ &\leq \sum_{i=0}^{n} \binom{n}{i} \sum_{j=0}^{i} \binom{i^2}{j^2} 2^{(j^2)} \\ &\leq \sum_{i=0}^{n} \binom{n}{i} \sum_{j=0}^{i^2} \binom{i^2}{j} 2^{j} = m(\Sigma_2) \\ m(\Sigma_2) &= \sum_{i=0}^{n} \binom{n}{i} \sum_{j=0}^{i^2} \binom{i^2}{j} 2^{j} \\ &= \sum_{i=0}^{n} \binom{n}{i} 3^{(i^2)} \\ m(\Sigma_3) &= \sum_{i=0}^{n} \binom{n}{i} 2^{(i^3)} \end{aligned}$$

The result follows from the observation:

$$n > 1 \Rightarrow 2^{(n^3)} > 3^{(n^2)}$$

The populatability of schemata $\{N_p\}_{p \geq 0}$ grows extremely fast. □

Lemma 6.

$$m(N_p) = \sum_{i=0}^{n} \binom{n}{i} 2^{(i^p)}$$

Lemma 7.

$$m(N_p) * m(N_q) \leq m(N_{p+q})$$

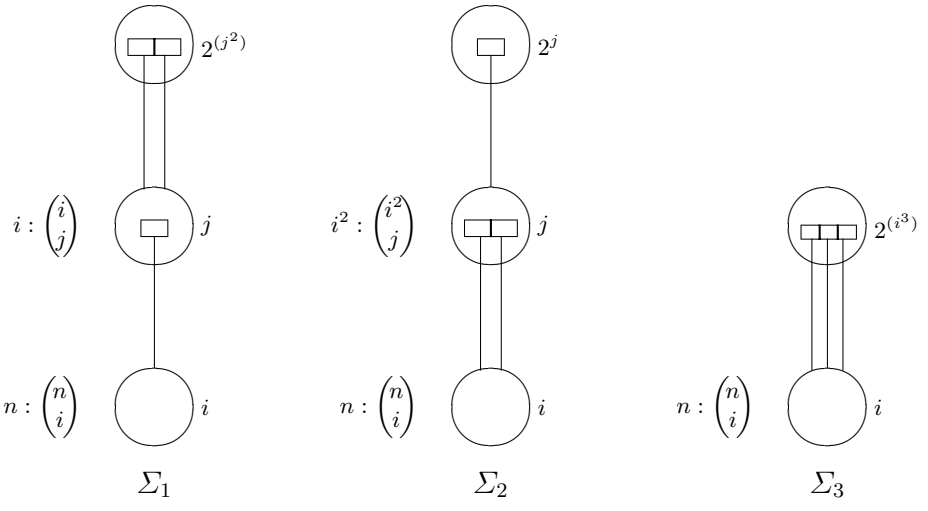


Fig. 7. Transformation steps

From theorem 1 we conclude that ER, NIAM and ORM are finitely bounded by their underlying domains.

4 Conclusions

In this paper we introduced some fundamental notions for the expression and comparison of the expressive power of conceptual schemata. In practise this will not be very helpful. However, by gaining a deeper understanding of the basic limitations of modeling techniques, we may be better equipped to improve state-of-the-art techniques.

Acknowledgements

The authors wish to thank Arthur ter Hofstede for his involvement in this paper.

References

[BHW91] P. van Bommel, A.H.M. ter Hofstede, and Th.P. van der Weide. Semantics and verification of object-role models. *Information Systems*, 16(5):471–495, October 1991.

[Bor78] S.A. Borkin. Data Model Equivalence. In *Proceedings of the Fourth International Conference on Very Large Data Bases*, pages 526–534, 1978.

[CAB+72] J.N. Crossley, C.J. Ash, C.J. Brickhill, J.C. Stillwell, and N.H. Williams. *What is mathematical logic?* Oxford University Press, Oxford, United Kingdom, 1972.

[HPW93] A.H.M. ter Hofstede, H.A. (Erik) Proper, and Th.P. van der Weide. Formal definition of a conceptual language for the description and manipulation of information models. *Information Systems*, 18(7):489–523, October 1993.

[ISO87] *Information processing systems – Concepts and Terminology for the Conceptual Schema and the Information Base*, 1987. ISO/TR 9007:1987. `http://www.iso.org`

PhDS 2005 PC Co-chairs' Message

With this Symposium for PhD students associated with the "On The Move Federated Conferences" we present the second edition of an event stimulating PhD students to summarise and present their results on an international forum. The Symposium supports PhD students in their research by offering an internationally highly reputed publication channel, namely the Springer LNCS proceedings of the OTM workshops, and by providing an opportunity to gain ample feedback from prominent professors. More specifically, students receive general advice on making the most of their research environment, on how to focus their objectives, on how to establish their methods, and on how to adequately present their results. These eminent professors thus help establishing a dialogue setting of scientific interrogation, discussion, reflexion and guidance between students and mentors. For taking up this challenging task, we would like to vividly thank our team of accompanying professors, namely:

- Domenico Beneventano, DataBase Group Dipartimento di Ingegneria dell' Informazione, Universitá di Modena e Reggio Emilia, Italy.
- Jaime Delgado, Distributed Multimedia Applications Group, Universitat Pompeu Fabra, Spain.
- Jan Dietz, Department of Information Systems, Technical University of Delft, The Netherlands.

In addition, we are much indebted to Werner Nutt, currently moving to the Free University of Bolzano, who, although he could not participate in the Symposium, was willing to take part in the review process. Finally, we also want to acknowledge the general OTM chairs who strongly stimulate the participation of PhD Students to this Symposium by offering them a publication channel such as Springer LNCS and an overall conference access at a much reduced rate.

We received 19 submissions by PhD students from 11 different countries, whereby each paper has been evaluated by three reviewers. 7 of these papers were accepted for publication.

We cordially invite the participants of the OTM conference and the readers of these proceedings to subsequently discover the next generation of promising scientists in our research field and get acquainted with their research topics.

Best regards from the OTM05 PhD Student Symposium program chairs.

August 2005

Antonia Albani, University of Augsburg
Peter Spyns, Vrije Universiteit Brussel
Johannes Maria Zaha, University of Augsburg
(PhDS'05 Program Committee Co-Chairs)

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, p. 740, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Accelerating Distributed New Product Development by Exploiting Information and Communication Technology

Darius Khodawandi

Delft University of Technology,
Chair of Information Systems,
PO Box 5031, 2600 GA Delft, The Netherlands
Darius.Khodawandi@darkor.de

Abstract. Enterprises are increasingly under pressure through globalization of markets, rapid advances in technology, and increasing customer expectations. In order to create and sustain a competitive advantage in this environment, adaptation of value generating activities is required. One of the value generating activities and the focus of the Ph.D. thesis is new product development (NPD). Especially the duration of NPD is a central success factor, as it has direct implications on its profitability and also on the strategic position of an enterprise. Efforts to accelerate NPD typically focus on increasing efficiency by exploiting potentials in the design of the product, the organizational structures, or the development process. Utilization of information and communication technology (ICT) for this purpose is dominated by support for actual engineering activities e.g. through computer-aided systems (CAx). The acceleration of NPD related processes supported by ICT especially beyond the boundaries of a single enterprise has received much less widespread acceptance and utilization so far. The specific question to be answered by the Ph.D. thesis in this context is therefore how to accelerate distributed NPD by exploiting ICT. The research results will be used for further development of existing software applications supporting NPD.

1 Introduction

Enterprises are increasingly pressured to compete in an international market in order to create and sustain a competitive advantage. The globalization of markets and the growth of electronic commerce have enabled enterprises to gain access to remote market places, leading to this increased international competition. Higher competition requires enterprises with an urge to deliver superior results to adjust value generating activities to those new international levels and on top exploit these market and technology trends to convert them into their own economic benefit (e.g., by redistributing business activities globally). In recent years e.g., production cost has become a major issue for the producing industry because of the excessive availability of labor at extremely low cost in China. At the same time products are becoming more technologically sophisticated, so integrating those rapidly advancing technologies into products becomes increasingly complex as well. Customers too are becoming more demanding in their preferences and expectations of a product. The result is that enterprises are

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 741 – 750, 2005.
© Springer-Verlag Berlin Heidelberg 2005

increasing their selection of products and at the same time experience a reduced product lifetime in the market (e.g., mobile phones or automobiles). In this context, enterprises are pressured to accelerate new product development (NPD). The research question addresses exactly this issue.

The remaining paper is organized as follows: section 2 deals with product development, specifically addressing the issues of NPD duration, distributed NPD as well as ICT in NPD. Section 3 defines the research question and provides an overview of current knowledge, preliminary ideas, the research methodology, and the current status of research. A summary of this paper is given in section 4.

2 New Product Development

NPD has been defined to include the activities beginning with the perception of a market opportunity and ending with the production, sale, and delivery of a product [9]. This puts a more interdisciplinary scope and focus to NPD than during its initial appearance in research and development (R&D) as well as engineering management literatures of the 60ies and early 70ies. Since then, NPD has emphasized different disciplines such as marketing, organizational theories, or strategy over time [3].

2.1 The Time Dimension in NPD

The concept of time-based competition [11] was the first to emphasize the importance of time as a central element of strategic management and its effects on competitiveness. In product development processes, time-to-market management is also a central success factor [6]. Both the timing of beginning product development as well as timing of market entry are determinants of overall profitability. Between these two points in time lies the process of NPD. The duration of NPD therefore limits the flexibility of a manager deciding on the above mentioned timing issues. Thus, it is an objective to accelerate NPD in order to provide managers the maximum possible flexibility in managing time-to-market.

Approaches to increase performance of NPD generally address the levers of efficiency and effectiveness. For example, by reducing the depth of value generation (i.e. buy instead of make) the duration of NPD can be reduced, but at the same time other challenges such as coordination of and contractual agreements with suppliers will arise. Another approach is to reduce the actual development workload by e.g. reducing product complexity, its variants, or features [5].

Under the aspects of strategic management and innovation, enterprise ability to introduce a product first in the market is directly associated with an innovator strategy. But not only innovators are dependent on a NPD duration that beats the industry. Any followers also have the strategic pressure to convert a management decision into a final product in the quickest possible time. And also from a cash-flow perspective, the less time product development takes, the sooner payback-time and therefore break-even will arrive, as shown in the following illustration:

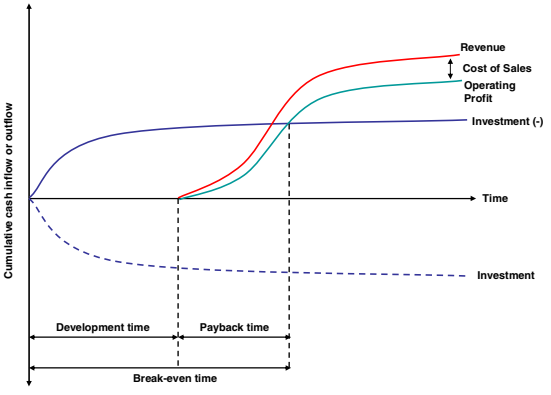


Fig. 1. Cash flows in the context of product development

Despite the advantages of speeding up product development, the downsides should be considered too. Saving time in the wrong places can lead to faulty products, e.g. because the product was not tested thoroughly enough and thus errors are experienced by customers. At the worst these faults can even be a threat to human lives. By reducing the duration of NPD, the time to recognize and anticipate changed market signals is shortened also. It may occur that initial trends change slightly over time, and that a competitor turns this change exactly into his competitive advantage in product specifications. So generally speaking, when accelerating NPD enterprises need to take into account the effects on the quality of their products [7].

The automotive industry is a good example for accelerating NPD. Competition in the automotive industry is increasing in multiple dimensions, including e.g. design, engine performance, fuel economy, or customer demands toward a broad choice of models. Aside from competition in the actual product, competition in NPD of automobiles has also increased notably. All automotive producers worldwide have been trying to reduce the time required from a products model freeze (i.e. the point in time when the basic design of the car is fixed) to its start of production (SOP) with Japanese manufacturers targeting 11 months for 2005 as shown in figure 2 [18].

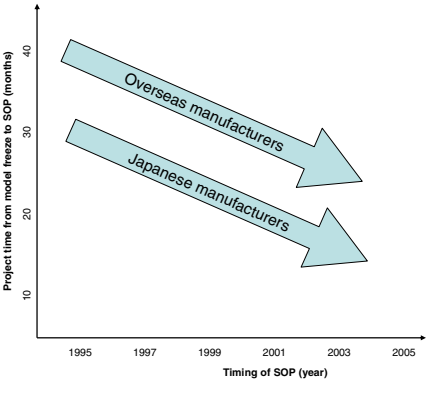


Fig. 2. Product development duration in the automotive industry

2.2 Acceleration of NPD

In order to accelerate NPD, [5] states that enterprises must increase either efficiency or effectiveness of NPD. On the one hand, this can be achieved by changing organizational structures and processes, or the product itself. To increase effectiveness, strategic R&D planning, market oriented product planning, management of the product portfolio, and strategic make or buy decisions for R&D are identified. The author argues that optimization of the concept of a product, the integration of suppliers in NPD, and management of specification changes will increase efficiency.

On the other hand, acceleration of NPD can be achieved by utilizing ICT. Support of NPD by ICT started with the development of computer-aided design (CAD) systems, which have evolved into more integrated and interoperable CAx-system suites. [4] further identifies rapid prototyping, virtual reality, digital mock-up, and product data management systems as major contributions to NPD by information technology. In the field of communication technologies, software for computer supported collaborative work (CSCW) aims at supporting cooperation in NPD. [7] argues that ICT specifically increases the speed of NPD as one of several impacts.

2.3 Distributed NPD

Intensive, frequent, and bi-directional communication and information sharing is a strategy for a successful and competitive NPD [12]. But NPD is typically not a process within a single enterprise or within an organizational unit of that enterprise. Instead, the activities of the process are distributed among multiple organizational and functional units and a network of enterprises all over the world. So, this strategy proves to be hard to implement. The distribution of value generating activities of NPD occurs for multiple reasons which are also motivated by the urge to accelerate NPD.

A major factor is said to be found in the continuing focus of enterprises on their core competencies [10]. In the case of the automotive industry e.g., the supplier network of a single original equipment manufacturer counts several hundred suppliers. Such suppliers specialized on delivering modules will have the responsibility to manage and integrate a network of enterprises themselves.

As a second major factor, distribution of NPD activities is possible because ICT reduces the setup costs for collaboration, therefore facilitating cooperation and coordination across organizational boundaries [2]. But the mere possibility to deploy value generating activities in a network of developers does not guarantee successful operations. Even though many of the challenges involved in this shift towards a product development network cannot be overcome without the utilization of ICT [7], the interoperability of ICT from enterprises involved is a critical challenge at the same time. A just as important success factor is the consideration of organizational issues, since enterprises involved will typically differ in their organizational structures, business processes, and their domain specific ontologies.

3 Research Question

The preceding two sections have given an overview of what the addressed problem area is. It has become clear that the NPD activities within and across multiple enter-

prises pose a challenge towards setting up an integrated NPD process with a competitive duration. Optimization in an organizational dimension (e.g. through cross-functional teams or supplier integration) as well as through ICT supporting engineering activities (e.g. CAD) have been dominant approaches to increase efficiency and effectiveness of NPD. Support of interorganizational collaboration with a focus on the time dimension of NPD has not specifically been addressed so far though. Thus, the central question to be answered in the doctoral thesis is:

How can distributed new product development be accelerated by exploiting information and communication technology?

The research question is independent of the industry at this point in time, so NPD activities e.g. in software, airplane, or mobile phone development should all be considered in the course of research for the thesis. Along with this primary question come several secondary questions to be answered as well:

Which are activities of NPD that offer the most potential for acceleration by exploiting ICT? At first, it is important to identify the primary activities in NPD. These are e.g. the definition of the product, the development of a supplier network, actual product development, production process development, and finally ramp-up. Each of these activities will have different characteristics of tasks and supporting ICT. For example, tasks during definition will be highly creative and involve many participants from different functional units of an enterprise. ICT support will be required in the visualization process, or in the process of rational evaluation and selection of product alternatives based on objective criteria. For each of these activities, the potential for acceleration by exploiting ICT must be evaluated.

What are the transaction patterns of these activities? Within these activities, typical patterns of communication and collaboration of involved enterprises and actors must be identified and analyzed. These transactions may differ for each activity significantly, because e.g. uncertainty will cause more intensified communications.

Which modeling language is compatible to these interorganizational patterns? After having identified and analyzed those patterns in reality, one or more modeling languages must be identified that are capable of representing the relevant aspects of transactions in the course of NPD.

Which ICT can exploit these patterns best? With the modeled transaction patterns at hand, the next question is by which ICT they are best supported. The fit between an ICT and the specific transaction pattern will determine the usefulness and optimization potential in terms of the time dimension of NPD.

What is a generic process to accelerate NPD by exploiting ICT? Finally, the above described process should be integrated into a generic process model to describe the single steps in order to accelerate NPD by exploiting ICT. This process model may then be employed in different industries at the best.

3.1 Current Knowledge and Existing Solutions

On the one hand, literature concerning the acceleration of NPD exists without specific respect to ICT. [6] deals with management of time-to-market and identifies levers for optimization in the areas of operations, management, and supporting processes. ICT is

only covered very briefly in the course of optimization in supporting processes. [5] is concerned with general management of NPD. Efficiency and effectiveness are identified as two central success factors in NPD. ICT is not included as a specific lever for optimization. [8] identifies four issues, on which efficiency and effectiveness of NPD depend on. These are the product technology strategy, organizational context, teams, and tools, with the latter group including ICT tools.

On the other hand, literature exists concerning the utilization of ICT within NPD, but not specifically addressing the issue of ICT supporting management of NPD, but rather concerning the issue of ICT supporting engineering activities (i.e. CAx). [4] gives an overview of ICT supporting NPD, including CAD, rapid prototyping, virtual reality, digital mock-up, and product data management systems. Collaborational aspects are only treated very briefly though. [7] identifies seven areas where ICT can facilitate NPD: speed, productivity, collaboration, communication, and coordination, versatility, knowledge management, decision quality, and product quality. [1] addresses support of collaborative NPD but is limited to setting up a strategic management framework.

The Ph.D. thesis research question is putting more emphasis on the supporting and transforming potentials of exploiting ICT in an interorganizational context. Also following this direction, [3] identifies several important research issues along four dimensions: process management, project management, information and knowledge management, and collaboration and communication. The author also states that the research agendas of NPD and ICT respectively currently do not reflect the evolutionary stage of ICT adequately.

3.2 Contributions to the Problem Solution

For research, results from the Ph.D. thesis will contribute to enterprise ontologies used for modeling interorganizational collaboration. The notations of the models involved may be subject to modification or extension in order to be sufficient for answering the research question in the domain of NPD. For industry, a modeling language to describe NPD processes in a network of enterprises will be practically proven in order to specifically address the industry issue of accelerating NPD. To further address this issue, a software application will be further developed on the basis of the results of the Ph.D. thesis, thus including software components supporting e.g. process management, project management, information and knowledge management, and collaboration and communication on an interorganizational level in order to generate adhoc-like processes for NPD (see section 3.5).

Compared to solutions for optimizing NPD by use of ICT in general, the doctoral thesis will emphasize the industry requirement of reducing the duration of NPD and using ICT in an exploiting manner. Compared to solutions specifically for reduction of NPD duration, the focus on exploiting ICT will increase efficiency without directly affecting engineering activities. Through this approach, the product quality risks described in section 2 especially connected to the simplification of products or non sufficient testing are avoided.

3.3 Preliminary Ideas

The most potential for accelerating NPD by exploiting ICT is in optimization of inter-organizational collaboration within NPD processes. An important aspect hereby lies in the interoperability in an organizational and technological dimension. In accordance with the additional research questions, significant processes in the course of NPD must be analyzed in order to identify patterns of interaction and information involved. By creating generic patterns and clustering information requirements, especially this interoperability issue is anticipated. A focus is set on patterns which offer the most potential for acceleration by exploiting ICT. The following basic activities of NPD are the basis for further analysis [2, 12]:

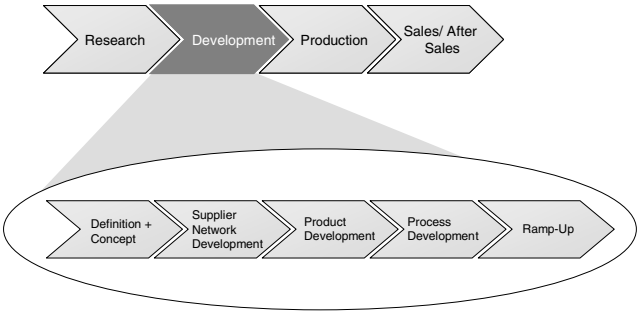


Fig. 3. Generic activities in NPD

At this point in time, ideas exist specifically for the supplier network development step. After product definition and concept creation, an enterprise will assess the situation of potential suppliers for the product. In the course of this step, enterprises will be occupied with the time-consuming steps of supplier identification, negotiation, evaluation, selection, and reaching a contractual agreement. An approach called strategic supply network development (SSND) [13,14] facilitates this step for generic supply chains, although the concept can also be applied to the NPD context.

A second source of potential for acceleration of NPD is seen in the ability to change the distribution of value generating activities (i.e. reconfiguration of the NPD network) more dynamically and in an adhoc-like manner. This requires utilization of ICT which support flexible configurations of value generating activities either within an enterprises boundaries or within a value network. The idea is to make distribution options in the value network available to the user at hand, hence including the option of distributing a specific activity in the course of process or project management. For example, an activity for developing the housing for a smart phone could have been executed within a product developing enterprise. By exploiting ICT, the enterprise could dynamically retrieve information about potential suppliers willing to deliver such a housing and possibly with pricing information and a delivery timeframe. By utilizing a generic transaction pattern including standardized information, the activity could be deployed to the external supplier and tracked electronically.

3.4 Research Methodology

Research will be conducted on the platform of an industry as well as an academic partner. On the research side, the partner involved is the Delft University of Technology, Netherlands. Within this University, the Faculty of Electrical Engineering, Mathematics and Computer Science holds the Department of Software Technology. Within this department, the section Software Engineering hosts a research program lead by Prof. Jan Dietz which is specifically dedicated to the topics of collaboration, interoperability, architecture, and ontologies, in short CIAO!. A central assumption of this program which will also be followed in the Ph.D. thesis is that ICT design should be human centered, thus the shaping element should be human interaction and collaboration during business processes.

The industry partner involved is a business consulting firm. It specializes on the optimization of processes along the value chain in general. The Ph.D. student is employed in a business unit dedicated to optimization of development processes in specific. The reduction of time required for NPD is a central objective in client projects.

An important component in the research methodology is practical observation (case research), because the work environment of the Ph.D. student is concerned with exactly the research question. Through the course of past and yet to come consulting projects specifically concerning NPD processes, a variety of industries and products to be developed can be covered. By applying enterprise ontologies from DEMO (Design & Engineering Methodology for Organizations) [15-17] to the business environments, insights about whether the applied methods are sufficient or require adaptation or extension are possible. In the course of consulting projects, research also is based on interviews with employees of client enterprises. The working environment of the Ph.D. student will additionally provide opportunity for feedback from experienced colleagues concerning NPD optimization.

3.5 Results Achieved So far and Next Steps

The Ph.D. student is in charge of reengineering a software tool used in NPD optimization projects with functionality in accordance with the process and project management dimensions from [3]. The tool enables enterprises to define an individual NPD process by using a certain structure and methodology. Included aspects of NPD are e.g. phases, milestones, activities, and responsibilities which are stored in a database. Activities cover the full scope of NPD as presented in the definition in section 2. Supported by the tool and based on the enterprise-specific NPD process, a project-specific NPD process is then customized for each NPD project. By visualizing the project instance of the NPD process and providing reports and work sheets, the tool supports process management and planning of the timeline and labor intensity of a NPD project. One of the effects for client enterprises from utilizing the application is the acceleration of NPD. Initially, the tool was limited to a single enterprises boundaries. Beginning in December 2004, a project was incepted to reengineer this tool into a web-enabled technology, thus at first providing the technological basis for functional expansion with the results of the doctoral thesis. The transformation of the tool resulted in a new architecture which permitted very flexible deployment. The architecture is shown in the following illustration and explained technically below:

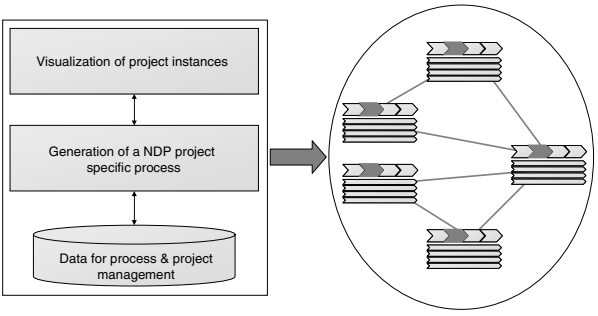


Fig. 4. Architecture of the current application and future deployment scenario

The application builds on a relational database. The application logic has bee implemented using the object-oriented programming language JAVA, and the execution environment is thus the Java Runtime Environment (JRE). Java Database Connectivity (JDBC) links the database and application logic. The user interface is browser-based and accesses the application logic through the Hypertext Transfer Protocol (HTTP). Each of these layers can be deployed on different hardware units, or all can be deployed on one. Thus, deployment is not limited to an enterprises boundaries, as long as network connectivity exists between the nodes of the deployment scenario.

Looking forward, the next steps are as follows. Future projects for NPD optimization will be utilized to answer the additional research questions and further evaluate the generic activities of NPD as well as the generic transaction patterns. In each of these projects, transactions will be modeled and may result in modifications or extensions to the applied modeling language(s). Another step in the future will be to determine existing software products that support NPD management, as well as their general strengths and weaknesses. Finally, the generic transaction patterns identified in the doctoral thesis will be specified as requirements for the functional expansion of the existing tool to add support for interorganizational NPD processes.

4 Summary

NPD is a key process for enterprises. In order to stay competitive, enterprises are thus required to accelerate NPD by increasing efficiency. The research question of the Ph.D. thesis addresses this problem and searches for solutions by exploiting ICT. A special focus is set on interorganizational collaboration during NPD. This is because increasingly distributed NPD activities raise requirements of communication and collaboration between entities involved, thus increasing the duration of NPD. In the course of research, generic transaction patterns will be identified, modeled, and implemented in applications to support NPD management. An application involved is currently being reengineered into a web-enabled technology. This will provide the basis for additional software components with functionality building on the research results. Next steps include the completion of the application reengineering, participation in client projects concerning optimization of NPD, and modeling of NPD transaction patterns using enterprise ontologies from DEMO.

References

1. Salminen, V., Yassine, A., Riitahuhta, A.: A Strategic Management Framework for Collaborative Product Development. In: Proceedings of the 4th International Conference on Engineering Design and Automation, Florida (2000)
2. Balakrisham, A., Kumara, S.R.T., Sundaresan, S.: Manufacturing in the Digital Age: Exploiting Information Technology for Product Realization. In: Information Systems Frontiers (1999) 25-50
3. Nambisan, S.: Information Systems as a Reference Discipline for New Product Development. In: MIS Quarterly (2003) 1-18
4. Tegel, O.: Information and Communication Technologies to Support Cooperation in the Product Development Process. In: Jürgens, U. (ed.): New Product Development and Production Networks. Springer Verlag, Berlin Heidelberg New York (2000) 389-406
5. Ohms, W.J.: Management des Produktentstehungsprozesses. Vahlen, München (1999)
6. Buchholz, W.: Time-to-Market-Management – Zeitorientierte Gestaltung von Produktinnovationsprozessen. Stuttgart (1996)
7. Ozer, M.: Information Technology and New Product Development. In: Industrial Marketing Management (2000) 387-396
8. Schilling, M.A., Hill, C.W.L.: Managing the new product development process: Strategic imperatives. In: Academy of Management Executive (1998) 67-81
9. Ulrich, K., Eppinger, S.D.: Product Design and Development. McGraw-Hill, New York (2000)
10. Hamel, G., Prahalad, C.K.: Competing for the future. Harvard Business School Press, Boston (1996)
11. Stalk, G., Hout, T.M.: Competing Against Time: How Time-based Competition is Reshaping Global Markets. Free Press, New York (1990)
12. Clark, Kim and Takahiro Fujimoto: Product Development Performance: Strategy, Organization, and Management in the World Auto Industry. Harvard Business School Press, Boston (1991)
13. Albani, A.; Keiblinger, A.; Turowski, K.; Winnewisser, C.: Dynamic Modelling of Strategic Supply Chains. In: Bauknecht, K.; Tjoa, A Min.; Quirchmayr, G. (eds.): E-Commerce and Web Technologies: 4th International Conference, EC-Web 2003 Prague, Czech Republic, September 2003, LNCS 2738 (2003) 403-413
14. Albani, A.; Winnewisser, C.; Turowski, K.: Dynamic Modelling of Demand Driven Value Networks. On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE. OTM Confederated International Conferences CoopIS, DOA, and ODBASE 2004, Agia Napa, Cyprus, October 2004, Proceedings, Part I, LNCS 3290 (2004) 408-421
15. Dietz, J.L.G., The Atoms, Molecules and Fibers of Organizations. In: Data and Knowledge Engineering (2003) 301-325
16. Dietz, J.L.G. Generic recurrent patterns in business processes. In: Business Process Management, LNCS 2687. Springer Verlag, Berlin Heidelberg New York (2003)
17. van Reijswoud, V.E., Mulder, J.B.F., Dietz, J.L.G.: Speech Act Based Business Process and Information Modeling with DEMO. In: Information Systems Journal (1999) 117-138
18. Internal report of industry partner consulting firm

Towards QoS-Awareness of Context-Aware Mobile Applications and Services

Katarzyna Wac

University of Geneva, CUI/OSG group,
24 rue General Dufour, 1211 Geneva 4, Switzerland,
University of Twente, EWI/ASNA group,
P.O.Box 217, 7500 AE Enschede, The Netherlands
Katarzyna.Wac@cui.unige.ch, k.e.wac@utwente.nl

Abstract. In our current connected wireless world, mobile devices are enabled to use various networking facilities. Although this enables mobile users to communicate *any time and any place*, it may also be very intrusive. There is a high need to manage the information stream a user receives on his/her mobile device. Context-awareness seems to be a promising way to manage this information stream and to provide the means to communicate at *the right time in the right way*.

Current context-aware applications benefit from the user context (e.g. location information), however, they do not consider the quality of service (QoS) offered by various networks (i.e. only best-effort QoS is considered). The research discussed in this paper focuses on a QoS- and context-aware service infrastructure supporting the development of mobile applications in a heterogeneous network environment. We argue that the use of context information helps to better capture the user's required QoS and improves the delivered QoS.

1 Introduction

The emergence of new wireless broadband networks and diverse miniaturized and personalized networked devices, give rise to variety of new mobile services in our daily life. Ultimately, these mobile services are executed as we move: in different places, at different time and under different conditions. Hence, these services get a continuously changing information flow from their execution environment. The management of this flow becomes vital for mobile services delivery. This means that a communication paradigm needs to shift from *any time and any place* into *the right time in the right way*, as the former may be very intrusive. Context-awareness is a promising way to manage the information flow, as context is any information that characterizes user's environment and situation, (e.g. location, time), and any object relevant to the interaction between the user and a mobile service [1].

For any mobile service the underlying communication is provided by heterogeneous network environment; consisting of wireless and wired networks. Each

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 751 – 760, 2005.
© Springer-Verlag Berlin Heidelberg 2005

network is responsible for a section of the 'end-to-end' communication path between a mobile user and application server placed in a service provider network (Figure 1[1]).

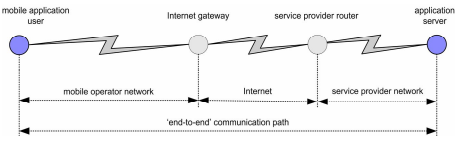


Fig. 1. An 'end-to-end' communication path for a mobile service delivery

Mobile connectivity, i.e., persistency of a wireless connection during the act of being mobile, and *quality of service* (QoS) *offered* by this connection, are critical factors for success of mobile service delivery. However, current rules for a connectivity choice are rather simple - a default wireless connection is chosen at a service-design time and assumptions are made regarding its offered QoS. Because a wireless link is usually a bottleneck in the end-to-end communication path, the assumptions regarding its offered QoS imply the assumption for the end-to-end offered QoS. Consequently, current mobile services are *delivered* with a best-effort (end-to-end) *quality*, and without consideration of the mobile user's *required QoS*.

The complication rises from the fact that nowadays various connectivity possibilities coexist offering different QoS. However, current mobile applications are unaware of that - the wireless network is chosen based on its availability, and not its offered QoS. Moreover, the mobile user's QoS requirements are not really considered or assumed to be static and attempted to be met with a best-effort service. Another complication is that the mobile application does not 'learn' from the end-to-end QoS experienced along the service delivery. This QoS information is not logged to be used further for QoS predictions, neither for a user himself nor for the other mobile users.

With respect to all given complications, we research on a QoS- and context-aware service infrastructure that supports the development of mobile applications in a heterogeneous network environment. We argue that mobile service infrastructure must not only be context-aware, but also QoS-aware; aware of the user's required QoS and QoS offered by various networks at the user's location and time. A wireless connection must be chosen with respect to what is required and what is offered. If necessary, service delivery should be adapted to what is offered (e.g. by means of changing the application protocol parameters). We argue that QoS- and context-awareness improves delivered QoS and that context-awareness helps to better capture the user context-depended QoS requirements.

Furthermore, we take the concept of QoS-awareness even broader and indicate that the actually delivered end-to-end QoS must be continuously logged by the mobile service infrastructure, and further used for QoS predictions. That leads to the proactive QoS-aware and context-aware infrastructure. We point necessity of development of a *QoS-context source*. It carries the responsibility of accumulation of logs on delivered end-to-end QoS from different mobile service users and provision of predictions of this QoS (i.e. along the particular trajectory traversed in a particular timeframe) to mobile applications. Hence, we aim in development of a 'route navigator' (i.e., outdoor and indoor GIS-based system) enriched with a QoS prediction for a particular user's trajectory.

[1] The mobile operator network comprises wireless access network and wired core network.

The following section gives an overview of the existing research related to the topic. Section 3 provides an explanation of our research context and section 4 - research trajectory within this context. The state of our thesis is discussed in section 5. The conclusion in section 6 summarizes our thesis goals and provides a critical evaluation of the work.

2 State-of-the-Art

There has already been a lot of research on QoS and context-awareness as separate topics, and this section presents only work the most important for us.

Projects like Equanet [3] developed a modeling-based performance evaluation method of the end-to-end QoS delivered to a mobile user over heterogeneous communication networks. In contrary, we consider measurements-based evaluation methods. Moreover, this project only focuses on VoIP and mobile web browsing mobile services, while we do not put constraints on the type of considered services. The CELLO project [4] concentrates on the location-based performance assessment of wireless communication infrastructures. Data regarding networks' performance is stored in a GIS system. However, as a performance indicator this project only considers signal strength and not the end-to-end QoS, as we do. Moreover, the overall goal of the project is to enhance the mobile operator network; the data is not used for mobile users, as we propose in our research. The publication of [5] provides a framework for network-aware applications and indicates that an application-level monitoring is one of the methods for application to be network-aware. However, this publication only considers the end-to-end bandwidth, and no other QoS parameters, as we propose. Similarly, [6] and [7] provide an idea on network resource awareness at the application level. Both indicate user context as necessary information for an application to adapt. However, both indicate the wireless access network and not end-to-end resources availability, as we do. Moreover, the mobile connectivity context source indicated in [7] is based on the single user's history of connectivity, and is used for a user himself to derive further context information. Hence, it will not be used for other users, as we indicate in our research.

Based on this short representation of the ongoing research, we define the innovative contribution of our thesis to the existing state of the art as the fact that we take the end-to-end QoS characteristics as context information and we introduce QoS-context source created based on information acquired from users and used for users.

3 Research Context

The context of our thesis is provided by the Dutch national Freeband AWARENESS project [8]. This project research an infrastructure that supports the development of context-aware and pro-active services and applications and validates it through prototyping in mobile healthcare (i.e., m-health) domain.

AWARENESS defines a three-layered architecture. The bottom layer is the *communication network infrastructure* layer, offering seamless mobile connectivity (e.g. 2.5G/3G/WLAN). This layer spans the 'end-to-end' communication path indicated in Figure 1. The middle layer is the *service infrastructure* layer providing an

execution environment for mobile services (e.g. service discovery and service context management functions). The top layer is the *application* layer offering *generic application services* like application-level context management and *domain specific services* like health tele-monitoring services, e.g., for the epilepsy, spasticity and chronic pain domains.

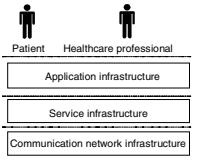


Fig.2. AWARENESS architecture

We position our research vertically across these three layers. We will incorporate the QoS requirements of service users and map them into system QoS requirements – into the requirements at the service infrastructure layer and then into the requirements at communication network infrastructure layer. To merge QoS-awareness and context-awareness, the context management function at the service infrastructure layer will be enriched with the management of QoS-context information. Moreover, we will develop a QoS-context source interacting with the service infrastructure layer.[2]

4 Research Trajectory

Our research trajectory consists of a few consecutive phases: 1) analysis of research-related concepts (to identify the current situation and its problems) as a basis for formulation of our research questions, 2) putting forward a hypothesis on a possible solution and 3) defending the hypothesis, eventually proving it to be a valid theory. Figure 3 presents our research trajectory cycle and particular actions taken in each of the phases. All phases may be repeated cyclically, if necessary.

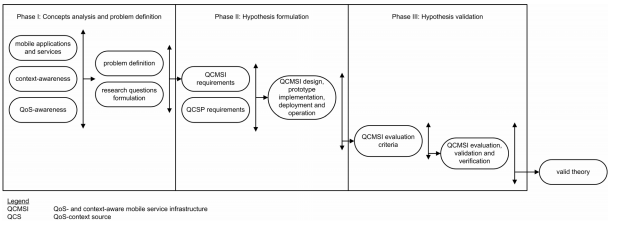


Fig. 3. Research trajectory

Phase I. In Phase I we start from the system related-concepts analysis and problem definition. We consider three main areas of literature review relevant for this research: a) mobile applications and services (e.g., current m-health applications), b) context-awareness (e.g., user/system context, current context-aware services and toolkits) and c) aspects of the QoS-awareness (e.g., what is QoS for mobile services, QoS management). This study results in a problem analysis of current context-aware service infrastructures and their QoS-(un)awareness.

Furthermore, Phase I aims in deriving specific research questions based on understanding the existing service infrastructure and its problems.

[2] I gratefully thank Dr Aart van Halteren for his supervision. This is an ongoing research in frame of the Dutch Freeband AWARENESS project (BSIK 03025), partially supported by the E-NEXT Network of Excellence (FP6 IST 506869).

Phase II. Following the research questions, in Phase II we will derive a hypothesis of a possible solution for the identified problem. As we stated earlier, we research on a QoS- and context-aware service infrastructure that supports the development of mobile applications in a heterogeneous network environment. Due to the nature of the problem, we derive our hypothesis based on the repeatable service lifecycle model [9], consisting of the following phases: a) requirements (i.e., set of features service must conform; service user and system[3] requirements), b) architectural design (describing the service organization in terms of structural elements, their composition, their interfaces and behavior), c) implementation (hardware, software and firmware used), d) deployment (service availability to users) and e) service operational phase (service maintenance).

Following this model, we will analyze requirements for: the QoS- and context-aware service infrastructure and for a QoS-context source. The former dictates the requirements for the latter. Based on these requirements, we will propose an architectural design for the QoS- and context-aware service infrastructure and its interface with QoS-context source. Following the AWARENESS project goals, prototype implementation and deployment of the system, will be done in m-health domain.

Phase III. The system prototype will aim in proving such whether the proposed solution fits the identified earlier problem (Phase I) and conforms the identified requirements (Phase II). Hence in Phase III we will attempt to defend (or refute!) our hypothesis. Firstly, we will derive evaluation criteria along which the prototyped system will be validated and verified. The evaluation will be executed with real m-health service users and will aim to prove that use of context information indeed helps better capture the user's required QoS and improves the delivered QoS. If we defend our hypothesis to be a theory, then we would like to indicate the utility of our solution for a mobile service in any application domain (e.g., commerce, entertainment).

5 Current Work

The research discussed in this paper started in November 2004 and consequently is in a starting phase. Some activities from Phase I are presented in the first sections of this paper and continued in this section together with the initial ideas for Phase II.

5.1 Research Questions

We research on context-aware mobile service infrastructure to enrich it with QoS-awareness. Moreover, we research on a QoS-context source. Therefore, we define the following research questions:

1) What are mobile user's end-to-end QoS requirements? How to translate them into the system requirements?
2) What end-to-end QoS context information is required at the context-aware service infrastructure level, how to get, and how to use it? How context may improve end-to-end QoS actually delivered to a mobile user?

[3] A system delivers a service; a service is an external observable behavior of the system.

3) What are the requirements for, and how to develop a QoS-context source created by users for users?
4) Based on the data aggregated in QoS-context source, what algorithms are used to predict end-to-end QoS along the mobile user's trajectory traversed in a given timeframe?

5.2 Epilepsy Tele-monitoring Scenario

The following future application scenario illustrates how an epileptic patient can benefit from context- and QoS-aware mobile health care service. The scenario (in boxed paragraphs), follow explanations of the technology we propose to support it.

> Sophie (28) is since four years an epileptic patient living in the Paris suburbs. Epilepsy is a neurological disorder, in which brain nerve cells release abnormal electrical impulses so-called seizures. Although the occurrence of a seizure is sudden and unexpected, she does not feel limited in her active life because she is treated under a continuous healthcare program; the Epilepsy Safety System (ESS) tele-monitors her health.

The ESS (Figure 4) is a distributed system, responsible for predicting, detecting and handling the occurrence of epileptic seizures. The ESS predicts seizure based on patient's vital signs[4]. Therefore, Sophie wears a Body Area Network (BAN), responsible for vital signs' data collection and monitoring. The BAN consists of sensors (to measure ECG, activity), a GPS module (to determine her location) and a Mobile Base Unit (MBU). An internal ad-hoc communication network (e.g. Bluetooth) connects the sensors, GPS module and the MBU. The MBU is a gateway between the internal and external (2.5G/3G/WLAN) communication networks and can be used to (locally) process the vital signs data. The MBU can be implemented as a Personal Digital Assistant (PDA) or a mobile phone. Sophie's vital signs are in a reliable manner, real-time available to be viewed in the healthcare centre. This constitutes an *m-health tele-monitoring service*.

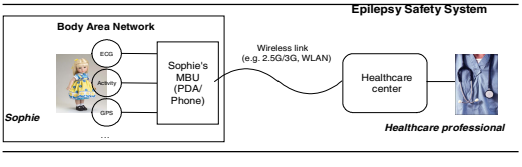


Fig. 4. Epilepsy Safety System architecture

To make Sophie's data available to her healthcare professional, the MBU connects to one of the external communication networks (e.g., WLAN/2.5G/3G) as available at her current location and time. To support Sophie's mobility, the MBU supports seamless handover between these networks.

Sophie's data have a maximum delay[5] defined by her doctor according to her current health state: 1 second for an emergency (i.e., seizure) case and 5 seconds

[4] According to the clinical research, an epilepsy seizure can be predicted in 80% of the cases at best 30 seconds before it, based on ECG and an increasing heart rate of a patient [8].

[5] Time elapsed from the moment the data is gathered from Sophie's body to the moment it is displayed to her doctor in the healthcare centre.

otherwise. This requirement is defined such that the healthcare centre provides medical assistance to Sophie in time. Besides the delay, the doctor defined which basic data (e.g. alarm and location information in case of a seizure) must always be made available to him and which are redundant (e.g., ECG signal). The delay and basic/redundant data definitions constitute very important quality of service (QoS) requirements for the tele-monitoring service. The kind and volume of data send to the healthcare centre depends on the capabilities of (i.e. QoS offered by) networks available at Sophie's current location and time. The MBU always selects the external communication network depending on its offered QoS.

In case of a seizure, depending on its severity, the ESS alarms Sophie, her healthcare centre and eventually Sophie's mother (if strong seizure). These activities take in total few seconds and they constitute an *m-health alarm service*.

Sophie is biking to the library in her village. Although she feels good, the ESS warns her of a possible seizure and triggers an alarm at the healthcare centre. She stops biking and sits on a bench near-by. Before she can ask for help, a seizure starts.

The 3G network is available at Sophie's location, so all sensor data together with her location information are continuously sent to the healthcare centre. Based on the ECG signals her doctor sees, he decides to intervene. When the ambulance reaches Sophie, medical professionals provide her with medical assistance and take her to the hospital. Sophie's doctor continues monitoring her while she is being transported. During the ride, the ambulance moves out of the 3G network range and the MBU transparently connects to a 2.5G network. Once Sophie arrives at the hospital, the MBU connects to WLAN and her ECG signals are automatically displayed in the emergency room.

When the MBU switches between different communication networks, the ESS adapts the signals it sends to Sophie's doctor. As result, the doctor will not see Sophie's ECG signals when the ambulance moves out of the 3G network coverage and the 2.5G communication network is used.

5.3 Requirements for QoS- and Context-Aware Mobile Service Infrastructure

To introduce QoS-awareness into a context-aware service infrastructure like AWARENESS, this infrastructure must meet the following requirements:

- QoS specification – the infrastructure must be able to handle user's end-to-end QoS specifications (e.g., data delay in our scenario).
- QoS mapping – the end-to-end QoS specifications must be mapped into the QoS requirements of the underlying communication network infrastructure.
- Assessment of the QoS-context information – the infrastructure must select the most suitable communication network based on the QoS offered by it and QoS actually required by a mobile user.
- QoS adaptation – when communication network handover occurs, the infrastructure must adapt service delivery to the QoS offered (section 5.5).

- Context adaptation – when context change occurs (e.g., an emergency), the QoS requirements may change. The infrastructure must incorporate context information when mapping QoS and adapting to QoS changes.
- QoS monitoring – the infrastructure must support real-time logging (i.e. measurements) of end-to-end QoS actually delivered to a mobile user.

5.4 Requirements for the QoS-Context Source

We indicate the QoS-context source as a source of the end-to-end QoS information aggregated over multiple mobile users. Following classification given in [10] we distinguish technology (i.e. network) oriented and user-oriented end-to-end QoS characteristics that this source aggregates. The network-oriented end-to-end QoS characteristic is its performance expressed in speed[6], accuracy[7] and dependability[8]. The user-oriented end-to-end QoS characteristics are a) perceived QoS, e.g., picture resolution, video rate/smoothness, audio quality, audio/video synchronization, b) service cost, e.g., per use or per unit cost and c) security level, e.g., authentication, confidentiality, integrity, non-repudiation.
The requirements for a QoS-context source are following:

- speed – real-time calculations and response to a mobile user
- accuracy – degree of correctness with which QoS-context is provided by source
- dependability – source availability and reliability
- scalability – support for a number of mobile users
- QoS-context information management
  - QoS-context information aggregation, pre-processing and inference
  - QoS prediction e.g., along given user trajectory in particular timeframe.

5.5 Incorporating Context

To support our argument that the use of context in a mobile service infrastructure improves the delivered QoS, subsection 5.5.1 presents how application protocol adaptation can benefit from QoS-context information when transporting user data over the 3G wireless networks. Section 5.5.2 shows how location-specific QoS-context information can further improve QoS adaptation and delivered QoS.

5.5.1 Application Protocol Adaptation

The mobile service infrastructure acquires from the QoS-context source information about the QoS offered by communication network infrastructures (WLAN/2.5G/3G) available at the user's current location and time. Service delivery must be adapted to the currently offered QoS. To illustrate it we use our knowledge about QoS offered by 3G networks (details in [11, 12]). Figure 5 shows the delay and goodput[9]

[6] Time interval used to transport data from a source to a destination [2].
[7] The degree of correctness with which a service is performed [2].
[8] The degree of certainty with which the service can be used regardless of speed or accuracy [2].
[9] Goodput - a throughput of a communication network infrastructure observed at the application layer.

characteristics of a 3G network as a function of the application protocol's packet size sent in an uplink[10]. Larger packet size results in a higher goodput but higher delays.

In a context-unaware situation, the application protocol designer maps user's QoS requirements to some fixed application protocol packet size; for a low-delay, a small size could be chosen and for an efficient (and cost-effective) use of the 3G network a larger packet size could be chosen. If we incorporate context, this packet-size could be adapted at the service run-time according to the context-dependent user's QoS requirements.

5.5.2 Location-and Time-Based QoS Adaptation

Another way of utilizing the QoS-context information is conduction of a seamless handover to the communication network offering better QoS than the currently used network. As service user is mobile, underlying networks, and their offered QoS change along the user's trajectory. Offered QoS fluctuates over time because of changing number of mobile users at a given location and their changing demands. Ideally, QoS-context source provides the real-time QoS-context information for the service infrastructure and QoS predictions along the given trajectory and for particular timeframe. The infrastructure must (proactively) recognize a QoS-context change.

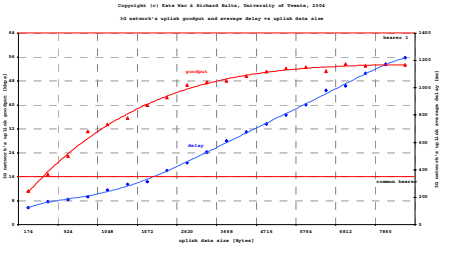


Fig. 5. Performance characteristics of 3G communication network infrastructures

In a context-unaware situation, the system designer maps service user's QoS requirements into QoS offered by some default network. If we incorporate QoS-context information in the infrastructure, the network choice could be done at the service run-time according to the context-dependent user's QoS requirements.

6 Conclusions

In this paper, we discuss our ongoing research on QoS- and context-aware service infrastructure that supports the development of mobile applications in a heterogeneous network environment. We define a research problem and we indicate possible research trajectory towards a valid-able solution. Hence, our research trajectory aims to prove that the use of context information plays a significant role in required QoS specification and it improves the delivered QoS. We indicate the necessity of QoS-context source providing the service infrastructure with predictions on QoS offered by the networks at the service user's particular position and time. Due to the nature of the identified problem, our validation technique is system prototyping. We will validate if a mobile user experiences improvements in delivered QoS while using QoS- and context-aware service infrastructure, comparing to the QoS-unaware one, offering only the best-effort service.

[10] From a mobile terminal to an application server.

We have already published one paper [13] disclosing our research topic and approving its relevance to the research community. We indicate the importance of our eventual findings and its applicability to improve QoS delivered by any mobile service. The target audience of our research is any mobile service provider, and as currently more and more applications and services go mobile, we indicate a high demand for our work. The beneficiaries of our work will be directly mobile users, experiencing their mobile services at the required quality. We identify the newness of our approach expanding beyond the standard QoS management framework comprising QoS contracts and QoS negotiation components. We propose user-driven approach, where user (and particularly a mobile application on user's behalf) will always have a choice amongst the underlying networks. This choice will be made with respect to user's QoS requirements.

Our research expands beyond the current telecom business model, where user is locked to one mobile operator. In our view, user needs to be able to make a decision, which network technology provided by which operator, is the most suitable to use. Moreover, by introducing the QoS-context source we further indicate user-empowerment; context information will be provided by users exclusively for users.

References

[1] D.Dey, *Providing Architectural Support for Context-Aware applications, PhD thesis, Georgia Institute of Technology*, USA, 2000
[2] ITU-T Recommendation I.350, *General aspects of Quality of Service and Network Performance in Digital Networks, including ISDNs*, March 1993
[3] Equanet project website, http://equanet.cs.utwente.nl
[4] CELLO project website, http://www.telecom.ntua.gr/cello
[5] J.Bolliger, T.Gross, *A framework based approach to the development of network aware applications*, IEEE Transactions on Software Engineering, vol. 24, no. 5, pp. 376 – 390, May 1998
[6] D.Chalmers, M.Sloman, *QoS and Context Awareness for Mobile Computing*, 1st Intl. Symposium on Handheld and Ubiquitous Computing (HUC'99), Karlsruhe, Germany, pp. 380 –382, September 1999
[7] J.Z.Sun, J.Sauvola, J.Riekki, *Application of Connectivity Information for Context Interpretation and Derivation*, 8th International Conference on Telecommunications (ConTEL 2005), Zagreb, Croatia
[8] M.Wegdam, *AWARENESS: A project on Context AWARE mobile NEtworks and ServiceS*, 14th Mobile & Wireless Communication Summit 2005, Germany
[9] I.Jacobson, G.Booch, J.Rumbaugh; *The Unified Software Development Process*; Addison-Wesley, 1999
[10] D.Chalmers, M.Sloman, *A Survey of Quality of Service in Mobile Computing Environments*, IEEE Communications Surveys, vol. 2, no. 2, 1999
[11] K.Wac, R.Bults et al., *Measurements-based performance evaluation of 3G wireless networks supporting m-health services*, Multimedia Computing and Networking 2005, Electronic Imaging Symposium, CA USA
[12] R.Bults, K.Wac et al. *Goodput analysis of 3G wireless networks supporting m-health services,* 8th International Conference on Telecommunications (ConTEL05), Zagreb, Croatia
[13] K.Wac, A.van Halteren, T.Broens, *Context-aware QoS provisioning in m-health service platform*, 11th Open European Summer School (EUNICE 2005), Colmenarejo, Spain.

Supporting the Developers of Context-Aware Mobile Telemedicine Applications*

Tom Broens

Centre for Telematics and Information Technology, ASNA group,
University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
t.h.f.broens@utwente.nl

Abstract. Telemedicine, which is defined as providing healthcare and sharing of medical knowledge over distance using telecommunication means, is a promising approach to improve and enhance the healthcare provisioning process. However, only recently, technology has evolved (i.e. miniaturization of high power mobile devices that can use high bandwidth mobile communication mechanisms) such that feasible advanced telemedicine applications can be developed. Current telemedicine systems offer proprietary solutions that are used in specific disease domains. For the acceptation, rapid development and introduction of novel and advanced telemedicine applications, there is a need for architectural mechanisms that supports developers in rapidly developing such telemedicine applications. The research discussed in this paper, focuses on the development of such mechanisms.

1 Introduction

Healthcare is intrinsic to human existence. Humanity has always been in need of solutions to various health related issues, such as childbirth and cure for diseases.

In the last decades, the introduction of ICT in this domain, was recognized as a valuable development to improve the healthcare provisioning process [1, 2]. The sub-domain of healthcare that uses ICT in its healthcare provisioning process is called *E-health* [3]. The evolution of ICT (e.g., growing processing power, mobile communication technologies) offers new possibilities to develop advanced e-health applications.

There are some major social-economic trends that stimulate and justify the need for E-health solutions [4]:

- *Patient-centric healthcare*: the offering of healthcare is shifting from offer- to demand-driven. The government does not primarily control the healthcare process anymore; the influence of the patient is increasing.
- *Cost savings and efficiency:* the society is aging. Currently, in Europe 16 to 18% of the population is over the age of 65. Estimations indicate that this will rise to 25%

* I would like to thank my supervisors M. van Sinderen and A. van Halteren for their contributions to my research and this paper. This work is part of the Freeband AWARENESS Project (http://awareness.freeband.nl). Freeband is sponsored by the Dutch government under contract BSIK 03025.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 761 – 770, 2005.
© Springer-Verlag Berlin Heidelberg 2005

in 2010 [4]. This increasing number of elderly results in an increasing number of potential healthcare consumers with a decreasing number of healthcare professionals.
- *Cross-domain integration:* to provide demand-driven healthcare and to keep costs in limit, the different domains in healthcare need to collaborate.

An example that supports the previous discussed issues is the current trend of extra-mural care compared to institutional care. Patients are treated as long as possible in their home environment instead of in care institutions. When they are hospitalized, the period of stay is minimized. This is both to save costs of hospitalization and to improve the patient's wellbeing.

The research areas in e-health are very diverse (electronic patient record, teleconsultation, patient management) and have their own specific issues. One interesting sub-domain of e-health, which is our particular focus of interest, is telemedicine.

Telemedicine is defined as providing healthcare and sharing of medical knowledge over distance using telecommunication means [1]. Although Telemedicine is already early recognized as a valuable improvement of the healthcare process, only recently technology has advanced in such a way that feasible advanced telemedicine applications can be developed (i.e. near real-time, high quality 24/7 telemedicine applications with relative low costs). On the one hand we see the rise of high bandwidth mobile communication mechanisms (e.g., GPRS, UMTS) and on the other hand we see the miniaturization of high power mobile devices [5]. However, still certain challenges limit or even block the development and introduction of these novel telemedicine applications. These issues are discussed in the following sections.

Section 2, gives an overview of the telemedicine domain and discuss the challenges in telemedicine by investigating current telemedicine applications. In section 3, the objectives for this research are presented. Section 4 discusses the approach taken to tackle the objectives. Section 5 gives the current status of this research. Section 6 discusses related work and in section 7 presents some conclusions and future work.

2 Overview of the Telemedicine Domain and Its Challenges

Telemedicine in itself consists of two sub-domains: (i) telemonitoring and (ii) teletreatment. Essentially, *telemonitoring* focuses on the unidirectional communication of vital signs from a patient to a healthcare professional. The *teletreatment* domain focuses on bidirectional communication, which also incorporates treatment data from the healthcare professional to the patient.

There are several fundamental problem domains that need attention for a successful introduction of telemedicine applications (and e-health in general) [4]:

- *Technology and Infrastructure:* what technologies and infrastructures are needed to develop efficient and useful telemedicine applications?
- *Political commitment:* To be able to create successful telemedicine application there has to be commitment by governments. For example, subsidising health innovation.

- *Legal & Ethics:* the exchange of private patient over public communication mechanisms is subject to legislation and ethical discussion.
- *Organization & Financing:* introduction of advanced telemedicine applications require drastic changes in the organization process of all stakeholders in the healthcare domain (e.g. professionals, government, insurance companies).

Although all problem areas need attention for an integrated and successful introduction of telemedicine applications, our research focuses on the technology and infrastructure issues which we discuss now in more detail.

We distinguished some key technological issues why the development of telemedicine applications is highly complex and costly (see [6]):

- *Limited generality and flexibility:* Currently available telemedicine equipment and infrastructures are proprietary solutions that do not provide generic capabilities for monitoring and treating arbitrary diseases. They focus on a single disease domain, like diabetics or hearth failure. This has several consequences for developing novel telemedicine applications:
  - *High learning curves:* Currently only proprietary telemedicine equipment and protocols are available which impose high learning, design and deployment curves.
  - *Costly:* Because of its specific use, telemedicine devices are very costly.
  - *Limited reuse:* Due to the focusing of current equipment and infrastructures on a specific disease domain (e.g. diabetic diseases), reusing applications for different domains is hard or even impossible and different equipment is needed. This is again costly.
- *Overload of information:* A side effect of using telemedicine applications is the fact that a huge amount of medical data is made available and has to be processed to be able to make good decision [7]. Furthermore, with the creation of health value-chains also other relevant healthcare information is coming available and has to be coupled with the primary healthcare data (e.g., vital signs). Without a mechanism to streamline these information flows, decision making based on this information becomes hard or even impossible. We claim that by using contextual information [8, 9] this streamlining can be done in an efficient manner.

3 Research Objectives, Questions and Scope

The previous section indicated some key technological issues that limit or even block the successful introduction and development of telemedicine applications. This leads us to the main objective of this research:

To develop architectural mechanisms that support application developers in developing context-aware mobile telemedicine applications.

This objective can be divided into several more focused sub-objectives:

- Propose an application framework that enables application developers to easily and rapidly develop context-aware mobile telemedicine applications.
- Distinguish generic and domain specific mobile telemedicine functions as part of the application framework.
- Analyze the impact of the proposed framework on the development process.

As discussed earlier, our main focus lies on the technological issues in telemedicine applications. This leads us to the following research question:

- What are the technological characteristics of telemedicine applications and how does this influence the development process?

We claim contextual information of all the entities involved in the healthcare process is needed to support integrated, efficient and patient-centric healthcare. This leads us to the following research questions:

- What contextual information is important in telemedicine applications and what are the consequences of introducing context in the development process?
- How can telemedicine applications benefit from contextual information?

The scope of this research is in the context of the Freeband AWARENESS project [10]. The AWARENESS project focuses on the research of an infrastructure that supports the development of context-aware and pro-active services and applications. AWARENESS validates this infrastructure through prototyping with mobile health applications. AWARENESS considers a three-layered architecture. The bottom layer of the architecture is the network infrastructure layer, offering seamless mobile connectivity. The middle layer is the service infrastructure layer that provides an execution environment for context-aware and pro-active services. It provides generic functionality like service life-cycle management, service discovery and security mechanisms. The top layer consists of the mobile health applications.

We position our research at the border of the two top layers. We have to incorporate the requirements and wishes of the telemedicine application developers and healthcare professionals and use the supporting functionality from the service infrastructure.

4 Approach

The approach taken in this research is divided into three main phases that are visualized, with their corresponding actions, in Figure 1:

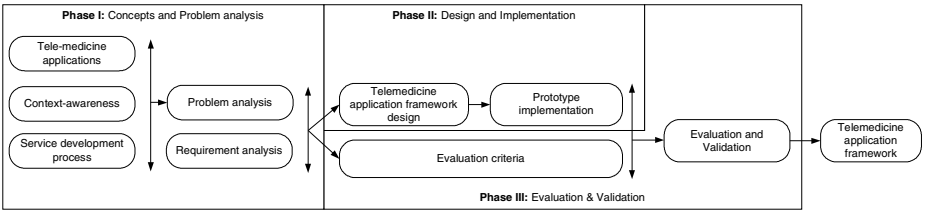


Fig. 1. Approach

Phase I: Consists of two major parts: (i) research on the state-of-the art in telemedicine applications, context-awareness and service development process (e.g., aspects important for software developers like development time, reusability,

flexibility and modularity, and (ii) research on the requirements of stakeholders in the healthcare domain. Analysis of both areas results in a problem analysis that identifies issues in the domain of current telemedicine applications.

Phase II: Design and implementation of the telemedicine application framework in a proof-of-concept prototype.

Phase III: Evaluation and validation of the developed framework based on criteria derived from research performed in phase I (e.g. development time, reusability). The validation is done by means of prototyping mobile telemedicine applications using our proof-of-concept application framework.

5 Current Status

The research discussed in this paper started in August 2004 and is therefore in its starting phase. Currently, we are working in three main areas of research that we discuss subsequently in the remainder of this section.

5.1 Application Framework for Mobile Telemedicine Applications

Although the benefits of telemedicine applications are widely recognized, implementations are scarce because of their current lack of flexibility, costly nature and the generation of an overload of information.
To improve the development time and reusability of these kinds of application, developers of telemedicine applications need mechanisms that provide them with flexible, extensible, portable, reliable, scalable and affordable ways to develop telemedicine applications. An *application framework* is an integrated set of software artifacts that collaborate to provide a reusable architecture for a family of related applications which has the potential to offer these characteristics [11].

In [12], we discuss our initial ideas on an application framework for mobile telemedicine applications that offers a generic execution environment. We consider an application as a set of collaborating application components deployed in the framework. The framework (see Figure 2) is positioned on top of the AWARENESS service infrastructure that was discussed in Section 3.

We distinguish three levels of generality within the functions a telemedicine application framework should provide. These levels relate to the different stakeholders in the development process of telemedicine applications:

- *Application container & Generic container functions* offer the generic execution environment for application components. These functionalities are developed by so called 'infrastructure developers'. This layer also offers the glue between the service infrastructure and the applications.
- *Domain specific functions* offer generic functionality for a particular domain. For instance, we distinguish two generic blocks of functionality (BANManagement and Signal processing), highly needed within the whole telemedicine domain. These functions can be reused by different applications. So called 'domain application developers' should develop these domain specific functions. Domain specific application developers

can be for instance a consortium of telemedicine organizations that have a similar interest.

- *Application components* represent the application for a particular end-user in a domain. Consider for instance a telemonitoring application for a particular doctor that wants to have the medical history of a patient specifically filtered to his need.

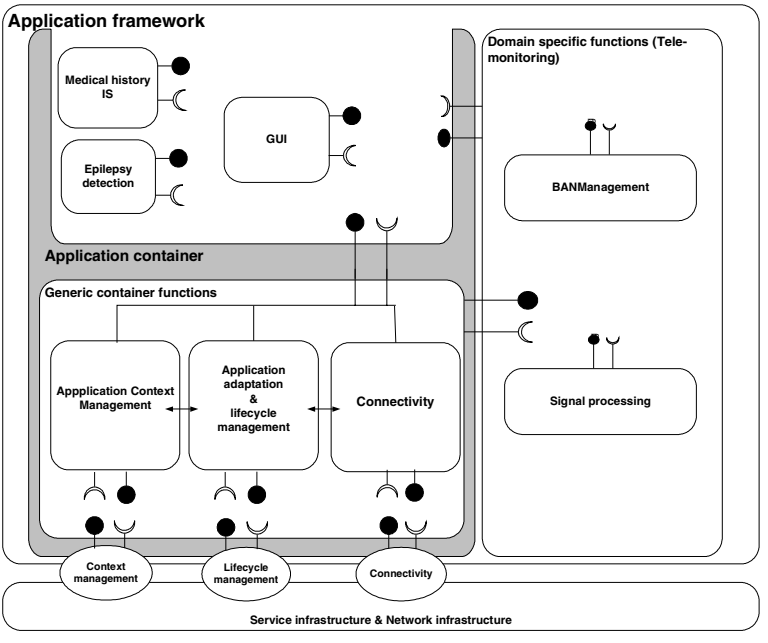


Fig. 2. Application framework for mobile telemedicine applications

The current design of the framework merely provides a functional decomposition. It does not (yet) elaborate on the functions and the mechanism needed to realize these functions except for BANManagement discussed in the next section. Designing the lacking functions are the next steps we want to take in the near future.

5.2 Modeling of Body Area Networks (BANManagement Component)

One of the domain specific functions mentioned in the application framework discussed in the previous section is the *BANManagement* component. In [13, 14], we discuss a realization of the BANManagement component. This component offers a toolbox for users and developers to manage a Body Area Network (BAN, i.e. network of body worn sensors/actuators and communication devices) of a particular patient. This includes the following activities:

- Configuration of the BAN;
- Instantiation of the BAN configuration;
- Reading vital signs and setting the parameters of actuators from the patient.

Again, several stakeholders are involved in the process of configuring, instantiating and reading data from a Body Area Network. We use the MDA (Model Driven Architecture) approach, based on the MOF Metadata Architecture [15], for the modeling of these Body Area Networks. See Figure 3 for an example of the process of managing a BAN for patient "Vic" using MDA meta-modeling.

A meta-model (M2 model) of a BAN is the basis for our toolbox. The model consists of *nodes* that are connected by *edges*. These nodes can be specialized as being sensors, actuators (leaf nodes) and aggregators, processors, stores, gateway (intermediate nodes).

Healthcare professionals, like specialists, can specify a BAN for a certain domain using this toolbox (e.g. epilepsy BAN (M1 model)). They create a kind of template. The healthcare specialist responsible for the primary care process, like nurses, can provide the specifics of a patients ban (e.g. Vic's BAN (M0 model)) by filling in the parameters of the Epilepsy M1 model (e.g. name of the patient, serial number of devices). The M0 model is used to read the specific sensors and set the actuators.

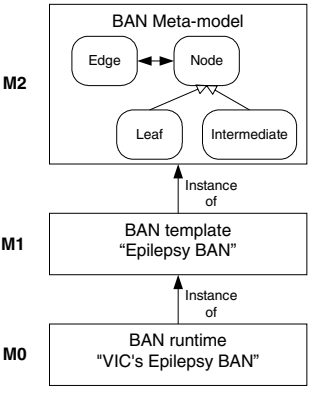


Fig. 3. MDA BAN modeling

Design of the discussed models is still current subject of research. In the future, we want to develop a prototype of the toolbox and subject it to case studies to evaluate it for usability by telemedicine application developers and healthcare professionals.

5.3 Prototyping Platform for Telemedicine Applications

Current BAN equipment is costly and inflexible. Therefore, we developed an telemedicine *prototyping platform* which can be applied for *prototyping* different problem domains (i.e. different diseases) using affordable, commonly available, off-the-shelf equipment.

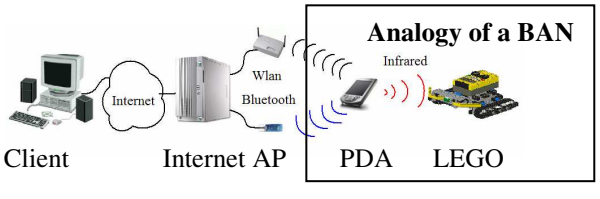


Fig. 4. Telemedicine prototyping platform overview

In this platform we use LEGO Mindstorms [16] technology and a PDA as analogy to a vital sign monitor with communication mechanisms. LEGO provides a so-called RCX unit that can handle three sensors (e.g., light, temperature) and three actuators (e.g., motors). Out of the box, LEGO transports this information with a proprietary

infrared protocol to a USB tower connected to a PC. We replace the PC with a PDA such that an arbitrary client from the internet can connect to the RCX (see Figure 4).

To enable this, we developed a daemon on the PDA with gateway functionality. This daemon bridges the gap between the client somewhere on the internet, and the RCX. Therefore, we implemented two protocols: (i) LEGO infrared protocol and (ii) light-weight TCP/IP based protocol for connection from the internet.

6 Related Work

The first research area we discuss in this paper is an application framework for telemedicine applications. Several types of application frameworks exist, offering functionality ranging from specific functionality like the Xerces XML parsing framework [17], more generic domain specific functionality like the Placelab positioning framework [18], to generic application support environments like CCM [19], J2EE (EJB) [20], J2ME (Midlets) [21], OSGi [22] and JADABS [23].

In general, we are mostly interested in the last category of application frameworks. Due to their big footprint, some of the frameworks (CCM, J2EE) are not suitable for use on mobile devices. Other frameworks, although suitable for mobile devices, are unchanged too limited for advanced telemedicine applications (J2ME Midlets). However, certain frameworks seem very interesting for realizing our telemedicine application framework (OSGi, JADABS). They are suitable for mobile devices and offer flexible mechanisms for application component deployment and communication.

The question that rises is how to use the mechanisms created by the discussed frameworks in our telemedicine application framework. Furthermore, the frameworks do not deal with the deployment of context-aware applications and the management of contextual information. However, several initiatives may help to compensate for the latter omission. Bardram [24] discusses a context-aware programming framework (JCAF) and Dey [25] discusses a toolkit for context management (Context Toolkit). These initiatives provide ways to acquire contextual information and transfer them to context-aware application. An interesting question is then how to integrate these context management mechanisms into a generic telemedicine framework.

Another aspect of our research is modeling of Body Area Networks. This is a relatively new research area, since only recently interest emerged in telemedicine applications using BANs. The IST Mobihealth project [26] takes a pragmatic approach in configuring a BAN which lacks flexibility for generic BANs. Laerhoven [27] discusses a XML based sensor model driven by similar issues as our BANManagement component. This model offers more flexibility but lacks mechanisms to define complex nodes in BAN like storage and processing nodes.

The final research area we discuss in this paper is a generic telemedicine prototyping platform. Again, this is a relative novel research area. There exist many disease specific platforms like Elite care [28] and Cardionet [29] which are not suitable for generic prototyping due to their specific nature. The Mobihealth project [26] and BSN networks [30] offer a more generic and extendible platform for telemedicine applications. However, they use specific sensor devices that are costly for prototyping activities. Furthermore, they lack the integration of contextual information.

7 Conclusion and Future Work

This paper discusses a PhD trajectory that focuses on the development of mechanisms to support application developers in developing context-aware mobile telemedicine applications. We indicate several reasons why there is a growing need for such applications. However, the currently deployed applications are too specific and costly to be useful in a generic way. This limits the development of such applications. Therefore, we research generic mechanisms to support the developer in rapidly developing telemedicine applications. We propose a context-aware application framework that separates generic functionality from domain specific and application specific functionality. This paper discusses one domain specific function that we want to offer to developers, namely BANManagement. Finally, this paper discusses a cost-efficient and flexible telemedicine prototyping platform that is going to be used for future prototyping efforts. Future directions we are exploring are:

- Extension of the application framework. This includes the identification of other relevant functional blocks and the realization of the distinguished blocks. Development of a useful component model and researching the application of available application frameworks like OSGi and JADABS for our framework.
- Prototyping and evaluation of the BANManagement component. This includes researching the characteristics of BAN and developing a demonstrator. This demonstrator is going to be evaluated by healthcare professionals and telemedicine application developers.
- Prototyping and evaluation of the application framework using the prototyping platform. This is the combination of the three discussed research areas in this paper. We want to deploy a prototype that incorporate the application framework and BANManagement component using the prototyping platform.

References

1. Pattichis, C., et al., *Wireless Telemedicine Systems: An overview.* IEEE Antenna 's and Propagation Magazine, 2002. **44**(2): p. 143-153.
2. Berg, M., *Patient care information systems and health care work: a socialtechnical approach.* International Journal of Medical Informatics, 1999. **55**: p. 87-101.
3. Oh, H., et al., *What is eHealth (3): A Systematic Review of Published Definitions.* Journal of Medical Internet Research, 2005. **7**(1).
4. The Telemedicine Alliance, *Telemedicine 2010: Visions for a Personal Medical Network.* 2004.
5. Marsh, A., *3G Medicine - The Integration of Technologies*, in *ICCS'02*. 2002.
6. Broens, T. and R. Huis in't Veld, *Tele-monitoring and Tele-treatment applications: problems and the awareness approach*, in *Freeband AWARENESS D4.5. forthcoming*. 2005.
7. Boulos, M., *Location-based health information services: a new paradigm in personalised information delivery.* International Journal of Health Geographics, 2003. **2**(2).

8. Zhang, D., Z. Yu, and C. Chin, *Context-Aware Infrastructure for Personalized Healthcare*, in *International Workshop on Personalized Health*. 2004: Belfast, Northern Ireland.
9. Dey, A., *Providing Architectural Support for Context-Aware applications*. 2000, PhD thesis, Georgia Institute of Technology.
10. Wegdam, M., *AWARENESS: A project on Context AWARE mobile NEtworks and ServiceS*, in *14th Mobile & Wireless Communication Summit*. 2005: Dresden, Germany.
11. Schmidt, D., A. Gokhale, and B. Natarajan, *Leveraging Application Frameworks.* ACM Virtual Machines, 2004. **2**(5).
12. Broens, T., et al., *Towards an application framework for context-aware m-health applications*, in *EUNICE: Networked Applications (EUNICE'05)*. 2005, ISBN: 84-89315-43-4: Madrid, Spain.
13. Broens, T., et al., *Context-aware application components*, in *Freeband AWARENESS D4.2*. 2004.
14. Jones, V., A. Rensink, and E. Brinksma, *Modelling mobile health systems: an application of augmented MDA for the extended healthcare enterprise*, in *EDOC 2005*. 2005: Enschede, the Netherlands.
15. OMG, *Meta Object Facility (MOF) Specification 1.4*, Series, 2002, Available from: http://www.omg.org/docs/formal/02-04-03.pdf.
16. LEGO Mindstorms, *LEGO.com Mindstorms Home*, Series, 2005, Available from: http://mindstorms.lego.com/eng/default.asp.
17. Apache XML project, *Xerces2 Java*, Series, 2005, Available from: http://xml.apache.org/xerces2-j/.
18. LaMarca, A., et al., *Place Lab: Device Positioning Using Radio Beacons in the Wild*, in *Pervasive Computing 2005*. 2005: Munchen, Germany.
19. OMG, *Corba Component Model version 3.0*, Series, 2002, Available from: http://www.omg.org/technology/documents/formal/components.htm.
20. Sun, *JAVA 2 Platform, Enterprise Edition (J2EE)*, Series, 2005, Available from: http://java.sun.com/j2ee/index.jsp.
21. Sun, *Java 2 Platform, Micro edition (J2ME)*. 2005.
22. OSGi Alliance, *The OSGi Service Platform - Dynamic services for networked devices*, Series, 2005, Available from: http://osgi.org.
23. JADABS, *JADABS*, Series, 2005, Available from: http://jadabs.berlios.de/.
24. Bardram, J., *The Java Context Awareness Framework (JCAF) - A Service Infrastructure and Programming Framework for Context-Aware Applications*, in *Pervasive Computing*. 2005: Munchen, Germany.
25. Dey, A., *The Context Toolkit: Aiding the Development of Context-Aware Applications*, in *Workshop on Software Engineering for Wearable and Pervasive Computing*. 2000: Limerick, Ireland.
26. van Halteren, A., *Mobihealth Generic BAN Platform*, Series, 2002, Available from: http://www.mobihealth.org/html/details/deliverables/pdf/MobiHealth_D2.2_final.pdf.
27. Laerhoven, K., M. Berchtold, and W. Gellersen, *Describing sensor data for pervasive prototyping and development*, in *Adjunct proceedings of Pervasive Computing 2005*. 2005: Munchen, Germany.
28. Standford, V., *Using Pervasive Computing to Elder Care.* IEEE Pervasive computing, 2002. **1**(1).
29. Ross, P., *Managing Care through the Air.* IEEE Spectrum, 2004: p. 26-31.
30. Lo, B., et al., *Body Sensor Network - A Wireless Sensor Platform for Pervasive Healthcare Monitoring*, in *Adjunct proceedings of Pervasive Computing 2005*. 2005: Munchen, Germany.

Multilingual Semantic Web Services

Frédéric Hallot

Chair INFO, Royal Military Academy, Brussels, Belgium
frederic.hallot@rma.ac.be
STARLab , Vrije Universiteit Brussels , Brussels, Belgium
frederic.hallot@vub.ac.be

Abstract. In this paper, an overview of the PhD thesis with the same name will be presented. After an introduction of the subject and aim of the thesis, a couple of research questions will be asked. A brief overview of the state of the art in the domain will be done and then some problems that arise with it will be considered. Although the monolingual problematic is quite general and recurrent in most ontologies design tools, this thesis will focus on the approach used in DOGMA Studio. Some new terms and acronyms will be introduced : "onternationalization", "concepton" and IMMO. Finally, some work issues for the thesis will be sketched then some conclusions will come before ending this document with some considerations about further problems that will follow the research.

1 Subject and Aim of the Thesis

The topic of the research is mainly based on the Semantic Web[BLHL01][AVH04] [DOS03][DFVH03][1]. The Semantic Web is a new type of Web oriented to software agents. It will not replace the actual web mainly used (browsed) by human users, but will coexist with it[BLHL01]. Because the Semantic Web will be mainly used by software agents, it will rely on Web Services [2].

Web Services are programs that run on Internet Servers and that implement a standard language that allows external programs to interact with them via the Internet, in order to let the Web Service compute a result that can be dependent on parameters send by the program who use the Web Service. The problem is often that different Web Services propose the same kind of data but present them differently. The description is slightly different (fname, f_name, First Name, ...) or the language used for describing the data are different (Naam, Nom, ...). Under these conditions it is very difficult to develop software agents that could query slightly different Web Services without putting in it a lot of complexity and also without going backward against one of the most important principles of computer sciences: data independence [3] [CD95]!

[1] see http://en.wikipedia.org/wiki/Semantic_web
[2] see http://www.w3.org/2002/ws/
[3] see http://www.webopedia.com/TERM/D/data_independence.html

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 771–779, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Therefore the need of the Semantic Web has become evident. The Semantic Web relies on ontologies[AJP04] [4]. Ontologies are a sort of Meta description of an abstract data structure. Mappings can be done from one ontology to slightly different concrete implementations. The writing of software agent becomes then easier because it can rely on the ontology as level of indirection and so query different types of Web Services.

A lot of research has already been done around the Semantic Web, and this one is at the very center of STARLab's research activity[5]. Until now most of the research in this field has been focused on only one language. The reason of this is that, even while staying in the scope of one unique language, a lot of complexities must be overcome. Those complexities come not only from synonymy of terms but mainly because of the semantic dependence of terms on contexts.

The aim of this thesis is to search how add the multilingual dimension in this field of research in the most efficient way. We hope that working with different languages will raise the level of language independence in semantic modeling. Language independence should be to ontologies what data independence has been to computers programs. This should help ontologies to become better, more pure semantic abstractions.

Anyway this multilingual dimension will be unavoidable in order to reach a Semantic Web that deserves its name!

2 Research Questions

2.1 Are We Designing Ontologies at the Right Level of Abstraction?

2.2 Could a Good Approach of Multilingual Ontologies Raise Their Level of Abstraction?

3 State of the Art

From all the modeling tools that we could evaluate, ranging from database modeling tools to ontology modeling ones passing by software modeling tools[6], there is not a single one that doesn't use linguistic terms as labels in order to identify the elements of their models. This means that each time we want to make a model (database schema, ontology, UML diagram,...) we have to choose one (spoken) language as the design language. And if we want several linguistic versions of a model, diagram, ... , we have to make several copies of the original one and then translate all the labels.

Some solutions have been developed in the field of software development in order to externalize the translations of the labels outside the programs. But

[4] see http://en.wikipedia.org/wiki/Ontology

[5] http://starlab.vub.ac.be

[6] We don't want to cite any because the list would be much too long in order to be exhaustive, but we are convinced that most readers will agree with us according to their own experience.

these techniques are not applied to modeling tools, and they don't solve the monolingual problematic in ontologies modeling that we will discuss further.

We insist on the fact that this monolingual problematic is much broader than the single field of ontologies. But because the thesis is called Multilingual Semantic Web Services we will remain in the domain of ontologies and for simplification reasons we will strictly focus on the DOGMA approach in this research. Anyway, most conclusions that will be drawn should be extendable to quite all modeling tools.

We would like now to very quickly introduce the DOGMA philosophy [VDB04] in order to be able to point at what we consider to be the problem in this particular case.

The main philosophy behind DOGMA is called the Double Articulation [MJ05] and resides in the separation of the ontology base and the commitment layer(see Fig. 1).

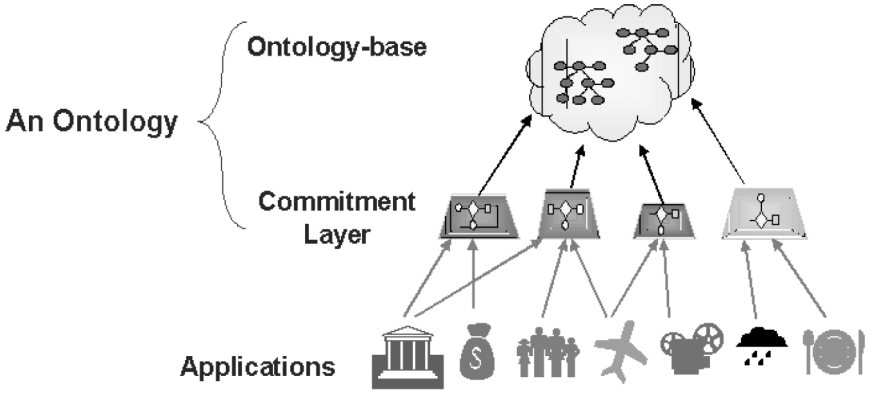


Fig. 1. the double articulation principle of the DOGMA philosophy (reproduced from [JM02])

This principle allows to keep ontology bases in a very basic expression. It is the responsibility of the commitments to make the link between the ontology base and the databases. Indeed, ontologies have been mainly developed to resolve interoperability problems between database applications developed separately in a common domain. Because of different designs (designers), the underlying data structures of similar database applications quite never match. To make them interoperable, there are two solutions:

- Develop a bridge between each pair of application, which implies $\frac{n*(n-1)}{2}$ mappings for n applications. The exponential character of this approach make it unaffordable.
- The second and only realistic way to obtain interoperability is to agree on a common description of the domain data and then make n mappings between the applications and the meta-data agreement. This agreed common meta description of the data is nothing else than the ontology base.

In the DOGMA studio philosophy, we need thus one ontology base and n commitments to make n applications interoperable.

The ontology base in the DOGMA philosophy is called the lexon base[VDB04], and an ontology consists thus of lexons.

The lexon formal definition was originally $< \gamma, t_1, r_1, r_2, t_2 >$[RM01], where γ stood for the context identifier, t_1 and t_2 were terms, and r_1 and r_2 corresponded respectively to the role and inverse role played by both terms in the relationship.

This first version of lexon was a bit to limited because didn't take into account any multilingual consideration. This definition has been revised and adapted [DBS03] to solve this problem. The new definition is $< \gamma, \zeta, t_1, r_1, r_2, t_2 >$ where ζ is the language identifier.

With this new definition appears the definition of meta-lexon $< C_1, R_1, R_2, C_2 >$. The lexon is now defined at the language level and the meta-lexon is defined at the conceptual level. Different lexons can now be linked with one meta-lexon. In this way it becomes possible to align different lexons just by linking them to a common meta-lexon that represent a concept.

With this new pair of definition the DOGMA philosophy seems now armed to face multilingual ontologies.

4 Current Problems Due to a Simplifying Mistake

In its probably most accepted definition [TG95] an ontology should be a shared agrement among users. It is clear that even if English is now the universal language of sciences, one should not forget that other spoken languages won't never disappear(or at least not before a very long time) and that the agrement among users will have to cross the border of the language used to define ontologies. This means that the multilingual dimension of ontologies will inexorably have to be taken into account, sooner or later. When the management of a company decide to postpone the multilingual dimension of an new application to be developed until the application has proved to be working efficiently in the company's language, it is more likely that the multilingual version will never come out, or at least not without a complete rewriting of the application[MK00]. Indeed, the multilingual dimension of applications implies some data structuring needs that should be directly tackled since the inception of the application.

These reasons should be enough to consider adding multilinguality to ontologies right away, before trying to solve the numerous problems that are still existing with the current monolingual approach. There is even a great probability that some of these problems will disappear just by the introduction of the multilingual dimension.

5 Implications

In this thesis, we would like to show that the monolingual approach to ontologies implies some problems and even mistakes against the essence of ontologies, and that a lot of energy is used in trying to find solutions to these implied problems.

Let's come back to the definition of a lexon in the Dogma approach. This definition is based on terms, in fact linguistics terms, that are coupled with the language used and the context in which they are used. Using linguistic terms as labels for defining lexons was according to us a huge mistake. Indeed the chosen label, that should only identify something, receives a semantic connotation, inherited by the possible significations of the chosen term. And even worse, because the chosen term has often several possible significations, each people reading the lexon has to decide which meaning he or she will give to this term! We think that this mistake is not a new one. In fact, it is probably this mistake that lead to most compatibility problems between databases inside a common domain of applications. Indeed "Name", "Lastname", "Surname", "Naam", "Nom", "Nom de famille",.... are several possible labels (sometimes in the same language) that can be chosen to denote a well know attribute in the context of Persons. Ferdinand de Saussure, considered as one of the father of modern linguistics, made a clear distinction between signifier and signified(see Fig. 2). In its approach, the signified means the concept about which one should agree and the signifier is the "acoustic image"[7] used to describe it.

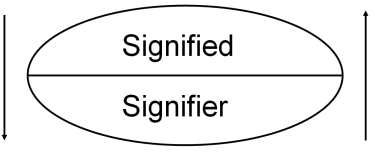


Fig. 2. Signifier and Signified (reproduced from [CJ97])

Some of the implied problems of the monolingual approach come from multinationalization[8] [MK00]. People writing multilingual applications are used to its two main aspects. Internationalization (I18n) and Multinationalization(M18n) [MK00]. Where Internationalization takes care of translating the messages of an application in different languages, Multinationalization takes the differences using a common language in different locations into account. Indeed, a unique sentence, in a common language can have a total different meaning in two different countries (regions) sharing the same language.

Even considering a unique language for defining an ontology using terms implies a lot of alignment problems. For example 'People has lastname' and 'people have surname' are considered to be two different lexons that have to be aligned. This is for us a conceptual mistake implied by the use of terms to define lexons. We would like here to refer to a very famous painting from René Magritte (A world famous Belgian painter) called "Ceci n'est pas une pipe" (This is not a pipe). For copyright reasons, it has not been possible to insert a copy of this painting here, but a research in Google will allow the reader to quickly

[7] In his theory[DS67] , Ferdinand de Saussure affirms that speaking is anterior to writing and that the study of linguistic should happen via the spoken language and not via the written language.

[8] Used here with a generic meaning.

get an idea of the picture on the Internet. This painting represent a pipe and this famous sentence in French "Ceci n'est pas une pipe". The idea behind the painting is that the painting is not a pipe, it is a representation of a pipe, which is totally different. In the same way considering 'people has lastname' as a part of an ontology is not correct. According to us, this is a possible representation of a part of an ontology. This confusion seems to be quite generalized in ontology engineering for now.

What is very strange is that this mistake was already solved a long time ago in the database field where several levels were defined in order to work with data. The physical level dealt with how the data would be physically stored(represented), the conceptual model dealt with the meaning of data and above that the possibility to define different views that could represent a same value in different ways. For example a price could be seen in USD or in EUR but denoting each time the same value. Unfortunately what had then been done for data representation, has never been done what concerns data description.

Ontologies were supposed to solve interoperability problems between similar databases, but by doing the same linguistic mistake, the same problems have occurred again and have occasioned a lot of research around ontology alignment.

6 Research Issues

The main goal of this research will focus on how to internationalize ontologies. Therefore, we would like to introduce a new word found only one time in google at this time of writing (very probably a type fault). This word is "onternationalization" and will be abbreviated as o18n. Several domain names have already been acquired by the author round this concept (www.o18n.com , .org , .net, .info), even if it will take some time before they will be effectively used.

We consider that the impedance mismatch between natural sentences and data structure descriptions, being relational schemas, XML files or ontology bases (lexons) is so huge that we can imagine an original system allowing to define all possible translations of terms inside predefined context, without having to cope with all the linguistic rules that govern sentence translation, or even sentence conversion to ontological constructions (lexons for example).

We think that the latest approach for modeling ontologies in the DOGMA philosophy is still not the best one, even though it seems to solve multilinguality problems for ontologies design.

- It's first default, is to keep on relying on terms for labeling the lexons (at linguistic level). As discussed before, we feel that this won't resolve all interpretation problems).
- Secondly (this problem was not considered yet in this paper) this force every different ontology designer to translate each time the same terms in the different languages that should be supported, and to add them in the lexon base that will grow consequently.
- Finally this new version has brought more complexity to the simplistic original version of lexons.

We think we should better revert to this original version, but with some adaptations. First of all, we would prefer to call the base element of an ontology a "concepton" in place of a lexon. Secondly we would like to definitively forget the terms and roles used in lexons. Our concepton would look like $< \gamma, CId_1, RId_1, RId_2, CId_2 >$. CId's meaning "Concept Identifier" and RId's "Role Identifier".

These identifiers would come from an International Multilingual Meta-Ontology (IMMO), where each identifier would be unique (by definition of identifier) and correspond to a concept (a role being after all also a kind of concept) that could be translated so many times as needed in every language (or even better locale). This would then mean that by defining one single concepton, we could obtain automatically a very large number of lexons, that would all be different but equivalent representations of a single concept.
This means also that translations of terms in the IMMO should only happen once, and then be reused at wish.

We can feel, that in this approach, there is still something missing. Indeed the ontology designer should forget labels for identifiers, which would make their task quite impossible.

We think that between the identifiers and the multiple translations, some sort of classification (taxonomy) should be defined. A lot of taxonomies already exists in different thesauri, and more research will be done in order to find an appropriate one. The advantage of this identifier approach is that even the taxonomy could be multilingual.

We further think of using the taxonomy as a kind of namespace, for example like in the Java programming language package construction. The taxonomy could then serve when designing ontologies to limit the number of visible Concept ID's. This means also that the taxonomy could eventually be used to denote the context in which a concept ID is defined. Some research will have to be done in order to investigate different existing solutions (Wordnet, Eurowordnet, Roget's Thesaurus,SIMPLE...) in order to see if they could be a good candidate to serve as base for the IMMO. A new construction could also be proposed, but it is not the aim of such a PhD thesis to develop a complete IMMO solution.

An extension of the actual implementation of DOGMA studio will be developed, in order to show how it could be possible to design ontologies based on CId's, without having to determine the labels anymore, but with the use of scrolling hints that will help to choose the right CId's and RID's, based on contextual enumeration of labels taken from the IMMO in a default design language.

Some work will also be done in order to expose the proposed IMMO structure as one or more Web Services, that would allow Semantic Web Services to become Multilingual Semantic Web Services!

Finally some research will eventually be done, to investigate the design of database (Relational, XML,...) using these CId's in the meta-data in place of the traditional linguistic labels. Using such databases, we could investigate for example how to query a same database using different (spoken) languages.

7 Conclusion

It is still a bit early to draw some conclusion at this time of the research, but we really hope that a good approach to multilingual ontologies will bring a positive answer to the research questions stated earlier in this document. We feel indeed intuitively that the level of abstraction of ontologies will raise consequently with this approach. As a consequence we can expect that the alignment problems due to synonymy, equivalent translations,... will disappear. This does not mean of course that all alignment problem will disappear. Indeed, even if using agreed identifiers to build ontologies, it will still happen that different builded ontologies in one same domain will still differ, but then it will be only because of semantical differences.

We also think, that even if the research is focused around the DOGMA Studio modeling tool, the IMMO could be integrated in quite all the existing modeling softwares in order to raise the level of abstraction at which the designers will work.

8 Further Problems

Of course one single PhD thesis will not solve this huge problem, with such an important matter. Some solutions will be envisaged, we hope maybe a good one. But even in this case, one can fear that it could take a very long time before it's adoption. Indeed onternationalization will only become effective if there is a very large consensus about it. In [KK04], the author explains that in order to resolve database mismatch problems, the Japanese Government envisages to use a "mappings first, schemas later" strategy where all databases designers in certain domains of application won't be able to arbitrarily define their data structure but should build them as subsets of an exhaustive domain definition imposed by the government. We believe that an IMMO will only make sense if it is an international initiative, supported by as much nations as possible. Building the infrastructure able to share online an International Multilingual Meta-Ontology, allowing to include all possible languages (taking even into account M18n versions) could only become possible with an official international institute mandated by, ideally all countries, maybe at UN level?

References

[AVH04] Antoniou G., van Harmelen F. - *A Semantic Web Primer* 2004, The MIT Press, ISBN 0-262-01210-3.

[CJ97] Cobley P. & Jansz L. - *Introducing Semiotics* 1997, Totem Books, ISBN 1-84046-073-3.

[BLHL01] Berners-Lee T., Hendler J., and Lassila O. - *The Semantic Web* May 2001, Scientific American.com

[DOS03] Daconta M.C., Obrst L.J., Smith K.T. - *The Semantic Web - A guide to the future of XML, Web Services and Knowledge Management* 2003, Wiley, ISBN 0-471-43257-1.

[CD95] Date C. - *An Introduction to Database Systems, 6th Ed.* 1995, Addison Wesley, ISBN 0-201-82458-2.

[DFVH03] Davies J., Fensel D., van Harmelen F. - *Towards the Semantic Web - Ontology-driven Knowledge Managment* 2003, Wiley, ISBN 0-470-84867-7.

[DBS03] De Bo J. & Spyns P.- *Extending the DOGMA framework in view of multilingual ontology integration. Technical Report 09* 2003, VUB - STAR Lab, Brussel.

[DS67] de Saussure F. - *Cours de linguistique générale* 1967, Grande Bibliothque Payot, ISBN 2-228-88942-3.

[TG95] Grubber T. - *Towards Principles for the Design of Ontologies Used for Knowledge Sharing* 1995, International Journal of Human-Computer studies, 43(5/6) : pp. 907-928.

[JM02] Jarrar M. & Meersman R. - *Formal Ontology Engineering in the DOGMA Aproach.* 2002 , in Meersman R., Tari Z. et al., (eds.),On the Move to Meaningful Internet Systems 2002, CoopIS, DOA, and ODBASE; Confederated International Conferences Coopis, DOA, and ODBASE 2002 Proceedings, LNCS 2519, Springer Verlag, pp. 1238-1254.

[MJ05] Jarrar M.- *Towards Methodological Principles for Ontology Engineering.* 2005, Phd Thesis, Vrije Universiteit Brussel, Faculty of Science.

[MK00] Kaplan M. - *Internationalization with Visual Basic* 2000, SAMS Publishing, ISBN 0-672-31977-2.

[KK04] Kuramitsu K. - *Managing Grid Schemas Globally.* 2004, 1st International Conference on Semantices of a Networked World, Semantics for Grid Databases, ICSNW PRE-PROCEEDINGS, Mokrane Bouzeghoub, Carole Goble, Vipul Kashyap and Stefano Spaccapietra (Eds.), pp.294-305.

[RM01] Meersman R. - *Reusing certain database design principles, methods and techniques for ontology theory, construction and methodology.* 2001, Technical report 01, STAR Lab, Brussel.

[AJP04] Pretorius A.J.- *Ontologies - Introduction and Overview. Starlab Internal Tutorial* 2004, VUB - STAR Lab, Brussel.

[PS04] Spyns P.- *Methods to be used in engineering ontologies. Technical Report 15* 2004, VUB - STAR Lab, Brussel.

[VDB04] Van den Broeck M. - *Towards Formal Foundations of DOGMA Ontology* 2004, Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science, VUB - STAR Lab, Brussels.

Improving Information Retrieval Effectiveness by Using Domain Knowledge Stored in Ontologies*

Gábor Nagypál

FZI Research Center for Information Technologies at the University of Karlsruhe, Haid-und-Neu-Str. 10–14, D-76131 Karlsruhe, Germany
nagypal@fzi.de

Abstract. The huge number of available documents on the Web makes finding relevant ones a challenging task. The quality of results that traditional full-text search engines provide is still not optimal for many types of user queries. Especially the vagueness of natural languages, abstract concepts, semantic relations and temporal issues are handled inadequately by full-text search. Ontologies and semantic metadata can provide a solution for these problems. This work examines how ontologies can be optimally exploited during the information retrieval process, and proposes a general framework which is based on ontology-supported semantic metadata generation and ontology-based query expansion. The framework can handle imperfect ontologies and metadata by combining results of simple heuristics, instead of relying on a "perfect" ontology. This allows integrating results from traditional full-text engines, and thus supports a gradual transition from classical full-text search engines to ontology-based ones.

1 Introduction

The huge number of available documents on the Web makes finding relevant ones a challenging task. Full-text search that is still the most popular form of search provided by the most used services such as Google, is very useful to retrieve documents which we have already seen (and therefore we know the exact keywords to search for), but it is normally not suitable to find not yet seen relevant documents for a specific topic.

The major reasons why purely text-based search fails to find some of the relevant documents are the following:

- *Vagueness of natural language*: synonyms, homographs and inflection of words can all fool algorithms which see search terms only as a sequence of characters.

* This work was partially funded by the VICODI (EU-IST-2001-37534) and DIP (no. FP6 - 507483) EU IST projects.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 780–789, 2005.
© Springer-Verlag Berlin Heidelberg 2005

- *High-level, vague concepts*: High-level, vaguely defined abstract concepts like the "Kosovo conflict", "Industrial Revolution" or the "Iraq War" are often not mentioned explicitly in relevant documents, therefore present search engines cannot find those documents.
- *Semantic relations*, like the *partOf* relation, cannot be exploited. For example, if users search for the European Union, they will not find relevant documents mentioning only Berlin or Germany.
- *Time* dimension: for handling time specifications, keyword matching is not adequate. If we search documents about the "XX. century" using exactly this phrase, relevant resources containing the character sequences like "1945" or "1956" will not be found by simple keyword matching.

Although most of the present systems can successfully handle various inflection forms of words using stemming algorithms, it seems that the lots of heuristics and ranking formulas using text-based statistics that were developed during classical IR research in the last decades [1] cannot master the other mentioned issues. One of the reasons is that term co-occurrence that is used by most statistical methods to measure the strength of the semantic relation between words, is not valid from a linguistic-semantical point of view [2].

Besides term co-occurrence-based statistics another way to improve search effectiveness is to incorporate background knowledge into the search process. The IR community concentrated so far on using background knowledge expressed in the form of thesauri. Thesauri define a set of standard terms that can be used to index and search a document collection (controlled vocabulary) and a set of linguistic relations between those terms, thus promise a solution for the vagueness of natural language, and partially for the problem of high-level concepts.

Unfortunately, while intuitively one would expect to see significant gains in retrieval effectiveness with the use of thesauri, experience shows that this is usually not true [3]. One of the major cause is the "noise" of thesaurus relations between thesaurus terms. Linguistic relations, such as synonyms are normally valid only between a specific meaning of two words, but thesauri represent those relations on a syntactic level, which usually results in false positives in the search result. Another big problem is that the manual creation of thesauri and the annotation of documents with thesaurus terms is very expensive. As a result, annotations often incomplete or erroneous, resulting in decreased search performance.

Ontologies form the basic infrastructure of the Semantic Web [4]. As "ontology" we consider any formalism with a well-defined mathematical interpretation which is capable at least to represent a subconcept taxonomy, concept instances and user-defined relations between concepts. Such formalisms allow a much more sophisticated representation of background knowledge than classical thesauri. They represent knowledge on the semantic level, i.e., they contain semantic entities (concepts, relations and instances) instead of simple words, which eliminates the mentioned "noise" from the relations. Moreover, they allow specifying custom semantic relations between entities, and also to store well-known facts and axioms about a knowledge domain (including temporal information).

This additional expression power allows the identification of the validity context of specific relations. E.g., while in the context of the "Napoleon invades Russia" event the "Napoleon – Russia" relation is valid, it does not hold in general.

Based on that, ontologies theoretically solve all of the mentioned problems of full-text search. Unfortunately, ontologies and semantic annotations using them are hardly ever perfect for the same reasons that were described at thesauri. Indeed, presently good quality ontologies and semantic annotations are a very scarce resource. This claim is based on both personal experiences during the VICODI project [5], and on our analysis of available ontologies and metadata on the present Web[1].

During the VICODI project an ontology-based web portal was developed which contained documents about European history[2]. A comprehensive ontology of European history was also developed. Although VICODI showed some of the potentials of an ontology-based information portal, the quality of the results were plagued by the lack of proper ontological information, due to the prohibitive cost of developing an ontology fully covering such a wide domain. The main lesson learned from the project is that it is very hard to switch from present full-text based information systems to semantic based ones in a one big step, but rather a gradual approach is needed, which combines the merits of statistical and ontological approaches, and thus provides a smooth transition between the two worlds.

In addition to the costs of ontology creation, another cause for imperfection is the limited expression power of ontology formalisms. Although they are much more powerful than thesauri, there are still many important aspects that cannot be modeled in present-day ontology languages. Therefore, imperfection in ontologies and metadata should be considered probably even in the long run, as the expression power of ontologies cannot be significantly raised without losing decidability of ontology reasoning.

The goal of this thesis is to examine and validate whether and how ontologies can help improving retrieval effectiveness in information systems, considering the inherent imperfection of ontology-based domain models and annotations.

This research builds on the results of VICODI, and therefore its major domain is also history. While history is a very interesting application domain from a theoretical point of view, it has also a strong practical relevance. After all, what is news today, will be history tomorrow. Therefore it is likely that the developed techniques will be directly exploitable in news portals on the Web. Indeed, we also plan to make some experiments with the IT-News domain.

The main contribution of this work is the demonstration of the utility of semantical information in an application domain of practical relevance without making unrealistic assumptions about the quality of ontologies and semantic metadata. This can provide a strong motivation for the creation of new ontologies which is a crucial step toward the Semantic Web vision.

[1] E.g. `http://www.daml.org/ontologies/`, `http://ontolingua.nici.kun.nl:5915/` and `http://protege.cim3.net/cgi-bin/wiki.pl?ProtegeOntologiesLibrary`

[2] Accessible from `http://eurohistory.net`

2 Approach

This research evaluates the following hypotheses:

1. Ontologies allow to store domain knowledge in a much more sophisticated form than thesauri. We therefore assume that by using ontologies in IR systems a significant gain in retrieval effectiveness can be measured.
2. The better (more precise) an ontology models the application domain, the more gain is achieved in retrieval effectiveness.
3. It is possible to diminish the negative effect of ontology imperfection on search results by combining different ontology-based heuristics during the search process which are immune against different kinds of ontology errors.
4. It is a well-known fact that there is a trade-off between algorithm complexity and performance. This insight is also true for ontologies: most of the ontology formalisms do not have tractable reasoning procedures. Still, our assumption is that by combining ontologies with traditional IR methods, it is possible to provide results with acceptable performance for real-world size document repositories.

These hypotheses are evaluated by implementing a prototype ontology-based IR system, and running experiments on a test collection. Gains in retrieval effectiveness in terms of classical IR measures such as precision and recall [1] are expected.

2.1 IR Process and Architecture

A schematic description of a usual IR process is shown on Fig. 1. Background knowledge stored in the form of ontologies can be used at practically every step of the process. For performance reasons, however, it does not seem to be feasible to use background information in the similarity measure used during matching and ranking, as it would be prohibitively expensive. Although, e.g., case-based reasoning systems apply domain-specific heuristics in their similarity measure [6], they operate on document collections which contain only several hundreds or thousand cases. We rather believe that it is possible to extend the query (and/or the document representation) syntactically based on the information stored in ontologies so that a simple, syntax-based similarity measure will yield semantically correct results (see also Hypothesis 4).

In this work, solutions are therefore provided for the issues of ontology-based query extension, ontology-supported query formulation and ontology-supported metadata generation (indexing). This leads to a conceptual system architecture (see Fig. 2) where the Ontology Manager component has a central role, and it is extensively used by the Indexer, Search Engine and GUI components[3].

2.2 Information Model

The information model defines how documents and the user query are represented in the system. The model used in this work is based on the model we

[3] The GUI component is responsible for supporting the user in query formulation.

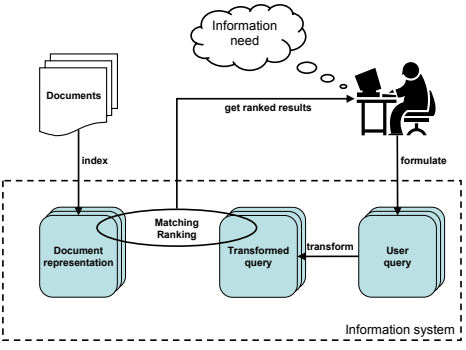


Fig. 1. IR process

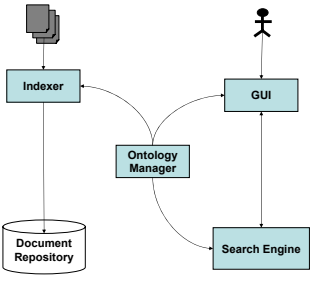


Fig. 2. Architecture

developed during the VICODI project, and represents the content of a resource as a weighted set of instances from a suitable domain ontology (the conceptual part) together with a weighted set temporal intervals (the temporal part). The representation of the conceptual part is practically identical with the information model used by classical IR engines built on the vector space model [1], with the difference that vector terms are ontology instances instead of words in a natural language. This has the advantage that retrieval algorithms, index structures, or even complete IR engine implementations can be reused with our model.

Time, as a continuous phenomenon has different characteristics than the discrete conceptual part of the information model. The first question according time is how to define similarity among weighted sets of time intervals. A possible solution which is being considered, is to use the "temporal vector space model", described in [7]. The main idea of the model is that if we choose a discrete time representation, the lowest level of granules can be viewed as "terms" and the vector space model is applicable also for the time dimension.

Another time-related problem is caused by some special properties of history as an application domain. Generally, it turned out that traditional time intervals are not suitable to represent historical time specifications because of uncertainty, vagueness and subjectivity. E.g., it is impossible to exactly specify the birth date of Stalin (there are two possible dates) or to define the precise starting date of the "Russian Revolution". To address these issues, a fuzzy temporal model together with a temporal algebra was proposed in [8]. Interestingly, recently a user study in the business news domain [7] further validates the claim that for some temporal specifications fuzzy time intervals are better suited than traditional time intervals (or a set of time intervals).

The previously mentioned "temporal vector space" model can be naturally extended to incorporate fuzzy temporal intervals. In this case the set of lowest level granules form the universe and fuzzy temporal intervals define fuzzy subsets of this universe.

A problem with the "temporal vector space" approach is the potentially huge number of time granules which are generated for big time intervals. E.g. to represent the existence time of concepts such as the "Middle Ages", potentially

many tens of thousand terms are needed if we use days as granules. This problem can be diminished by "granule switching", i.e. using bigger granules for queries and/or documents which define a wide time range as relevant. The intuition is that in those cases the information loss caused by the transformation will not distort relevance scores significantly.

In addition to the original VICODI model, our information model also contains a traditional bag of words representation of the document content (and query terms), as we see the ontology-supported IR heuristics as an extension of the traditional IR approaches and not as a replacement.

2.3 Query Formulation

VICODI experience showed that navigation in a full-fledged ontology is too complicated for most of the users. Therefore during query formulation we use the ontology only to disambiguate queries specified in textual form. E.g., if the users type "Napoleon" we provide them a list of Napoleons stored in the ontology (by running classical full-text search on ontology labels), and users only have to choose the proper term interpretation. This is much easier than finding the proper Napoleon instance starting from the ontology root. We also plan to make experiments with completely automatic disambiguation techniques like in [9] and in [10].

2.4 Ontology-Supported Query Expansion and Indexing

Recent research shows [11] that a combination of ranking results of simpler queries can yield a significantly better result than a monolithic query extension. This is probably because every algorithm has its own strengths and weaknesses and it is not possible to find "the optimal" method. Therefore, it is better to combine the results of different algorithms than just using only one (see also Hypothesis 3).

Motivated by these results our query process applies various ontology-based heuristics one-by-one to create separate queries which are executed independently using a traditional full-text engine. The ranked results are then combined together to form the final ranked result list (see Fig. 3). The combination of results is based on the belief network model [12] which allows the combination of various evidences using Bayesian inference. New types of ontology-based or statistical heuristics can be easily added to the system. If no ontological information is available, the system simply uses the "bag of words" part of the information model. Therefore the proposed search process supports a gradual transition between full-text and ontology-based IR systems.

2.5 Test Collection and Evaluation Strategy

A new test collection has to be built because unfortunately, presently no test collections are available that incorporate ontologies. The approach for building the test collection is the following: Wikipedia is reused as the basis[4], which is

[4] Available for download at `http://download.wikimedia.org/`

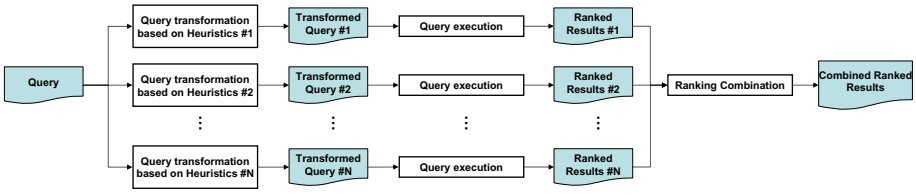


Fig. 3. The search process

refined by adding documents describing specific miniworlds, such as "World War II" or the "Iraq War". For these miniworlds (specific topics) facts are also added to the ontology and metadata is generated.

In other words, we follow the gradual approach also during test collection creation, as it is clearly not feasible to semantically annotate the whole Wikipedia for these experiments. After this process there should be miniworlds that are only partially or erroneously covered by semantic information, and there should be other topics which are not covered by the ontology at all. This is needed to test the robustness of the approach.

One advantage of this approach is that recall can be estimated well, as most of the relevant documents for the miniworlds are added manually to the test collection. Wikipedia can also contain relevant documents, of course. Those will be explored using the pooling method, which is also used on the TREC conferences (see e.g. [13]) to estimate the number of relevant documents in big test collections. Further, since Wikipedia contains more than 700000 documents, the performance of the system can be tested in a real-world situation.

Classical IR laboratory experiments will be conducted to get recall and precision figures. Various ontology-based heuristics will be switched on one-by-one, and a gradual gain in effectiveness is expected, which will hopefully validate Hypothesis 1 and partially Hypothesis 2. To further validate Hypothesis 2, fuzzy time information will be switched on and off to see if the simplification caused by using traditional crisp temporal intervals degrade retrieval effectiveness. To validate Hypothesis 3, the results of various heuristics will be integrated into one query extension step, and the retrieval figures will be compared with the other approach which combines the ranking results. Hypothesis 4 will be validated by comparing the response time of the new system with the response time of traditional full-text search.

2.6 Research Status and Implementation

The fuzzy time model for modeling temporal specifications in history is complete, presently we are working on tool support for defining such intervals in a user-friendly way. Tool support for ontology development is in place and it is improved continuously. The test collection for the evaluation is presently being created.

During the VICODI project a first prototype of the described IR system was developed [14], using crisp temporal intervals and a one-step query expansion. This system used the KAON1 system [15] for reasoning, which unfortunately

did not scale well. Based on the experiences with the VICODI system a new prototype is presently under implementation using the more powerful KAON2 system[5], fuzzy intervals and the ranking combination approach. The KAON2 system is used as an efficient disjunctive Datalog engine. Although KAON2 also supports OWL-DL reasoning, Datalog is used directly as the ontology implementation language because OWL-DL (and generally description logic) is not suitable for the application domain of history because some required reasoning patterns, such as temporally dependent transitive partOf relations on locations, are not supported. KAON2 also supports user-defined datatypes, which is crucial for implementing fuzzy time interval-based reasoning in the ontology.

As full-text retrieval engine, the Apache Lucene library[6] is used.

For the ontology-supported automatic semantic metadata generation, we use the GATE system[7], and develop GATE components which can exploit background information stored in our ontology. During the indexing process we also exploit traditional GATE components which are not ontology-aware, i.e. we follow the same gradual approach as during the query process.

3 State of the Art

It is well known fact in the field of IR that simple syntactical matching of the document and query representations do not always yield optimal results. A big body of literature exists where approaches using thesauri is described (see e.g. [1], Chapter 5 for an overview). This includes both automatically constructed thesauri based on statistical properties of the document collection or hand-crafted thesauri. As ontologies also codify background knowledge of a domain, lessons learned for thesauri are also relevant for ontologies.

Another related area is Information Extraction (IE) [16] that provides methods for extracting pieces of information from textual documents. Results of this research are useful for the indexing task in the IR process. Although usually a general claim is made by the IE community that semantic indexing (extracting semantic metadata from documents) provides better retrieval effectiveness than traditional full-text search, the emphasis of these systems is not retrieval but indexing. Probably therefore these claims are not yet validated, although because of the imperfection issues this claim is not trivial.

Recently, ontology-based information retrieval attracted a considerable interest. Most of the systems, however, concentrate on retrieving ontology instances, rather than documents [10,17]. The approaches of these works can be used, however, as part of our ontology-based heuristics to extend the IR query.

Probably the most relevant works to this research are the KIM system [18] and the work reported in [9]. They also define a general framework for ontology-supported document retrieval, and integrate full-text search with ontology-based methods. These systems, however, start with a semantic query in an ontology

[5] Available from `http://kaon2.semanticweb.org`
[6] `http://lucene.apache.org`
[7] `http://gate.ac.uk/`

query language and use the resulting instances to retrieve relevant documents. This is different from our approach where the ontology is rather used to extend a non-ontological query, which also contains semantic elements. This has the advantage, that our system can be also used for information filtering because the representation of documents and queries coincide.

KIM uses the retrieved ontology instances to initiate a traditional full-text search on documents where ontology annotations are embedded into the document text as terms. We also use a traditional search engine on the syntactic representation of the ontology elements, but we do not embed annotations to documents.

In the system of Vallet et al. documents are connected with ontology instances via weighted annotations, which is practically the same solution as our "bag of ontology instances" model. As they also include documents and annotation to the ontology, they can directly use the annotation weights to calculate semantic document relevance using the classical tf-idf formula, after executing the ontology-based query. They also execute a full-text search separately and combine its returned relevance weight with the result of the semantic query to diminish the effect of ontology imperfection. This is a similar, but simpler idea that we use when we combine the results of ontology-based heuristics.

None of the solutions handle the temporal dimension separately, and correspondingly they do not provide any support for fuzzy temporal intervals which is needed in some application domains such as history or business news.

4 Conclusion

This PhD work examines how background knowledge stored in ontologies and semantic metadata can be optimally exploited for the task of information retrieval. A special emphasis is placed on the issue of imperfect ontologies and metadata which is the reality on the present Web. History is used as application domain, which is very challenging from the temporal modeling perspective and allows evaluating the effect of ontology modeling simplifications on retrieval effectiveness.

During further work we validate that the proposed solution significantly improves retrieval effectiveness of information systems and thus provides a strong motivation for developing ontologies and semantic metadata. The gradual approach described allows a smooth transition from classical text-based systems to ontology-based ones.

References

1. Baeza-Yates, R.A., Ribeiro-Neto, B.A.: Modern Information Retrieval. ACM Press / Addison-Wesley (1999)
2. Kuropka, D.: Uselessness of simple co-occurrence measures for IF&IR – a linguistic point of view. In: Proceedings of the 8th International Conference on Business Information Systems, Poznan, Poland (2005)

3. Salton, G.: Another look at automatic text-retrieval systems. Commun. ACM **29** (1986) 648–656
4. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American **284** (2001) 34–43
5. Nagypál, G., Deswarte, R., Oosthoek, J.: Applying the Semantic Web – the VICODI experience in creating visual contextualization for history. Literary and Linguistic Computing (2005) to appear.
6. Lenz, M., Bartsch-Spörl, B., Burkhard, H.D., Wess, S., eds.: Case-Based Reasoning Technology: From Foundations to Applications. Volume 1400 of Lecture Notes in Computer Science. Springer (1998)
7. Kalczynski, P.J., Chou, A.: Temporal document retrieval model for business news archives. Information Processing & Management **41** (2005) 635–650
8. Nagypál, G., Motik, B.: A fuzzy model for representing uncertain, subjective, and vague temporal knowledge in ontologies. In: On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE, Springer (2003) 906 – 923
9. Vallet, D., Fernández, M., Castells, P.: An ontology-based information retrieval model. In: The Semantic Web: Research and Applications: Second European Semantic Web Conference, ESWC 2005, Springer (2005) 455–470
10. Rocha, C., Schwabe, D., Aragao, M.P.: A hybrid approach for searching in the semantic web. In: WWW '04: Proceedings of the 13th international conference on World Wide Web, ACM Press (2004) 374–383
11. Silveira, M.L., Ribeiro-Neto, B.: Concept-based ranking: a case study in the juridical domain. Information Processing & Management **40** (2004) 791–805
12. Ribeiro, B.A.N., Muntz, R.: A belief network model for IR. In: SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, ACM Press (1996) 253–260
13. Voorhees, E.M., Buckland, L.P., eds.: NIST Special Publication 500-261: The Thirteenth Text REtrieval Conference Proceedings (TREC 2004). (2004)
14. Surányi, G.M., Nagypál, G., Schmidt, A.: Intelligent retrieval of digital resources by exploiting their semantic context. In: On The Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE, Springer (2004) 705–723
15. Motik, B., Maedche, A., Volz, R.: A conceptual modeling approach for semantics-driven enterprise applications. In: Proc. 1st Int'l Conf. on Ontologies, Databases and Application of Semantics (ODBASE-2002). (2002)
16. Vlach, R., Kazakos, W.: Using common schemas for information extraction from heterogeneous web catalogs. In: Proceedings of the 7th East-European Conference on Advances in Databases and Informations System (ADBIS), Springer (2003) 118–132
17. Stojanovic, N., Studer, R., Stojanovic, L.: An approach for the ranking of query results in the semantic web. In: ISWC 2003. (2003) 500–516
18. Kiryakov, A., Popov, B., Terziev, I., Manov, D., Ognyanoff, D.: Semantic annotation, indexing, and retrieval. Journal of Web Semantics **2** (2005)

Top-k Skyline: A Unified Approach

Marlene Goncalves and María-Esther Vidal

Universidad Simón Bolívar,
Departamento de Computación,
Caracas, Venezuela
{mgoncalves, mvidal}@usb.ve

Abstract. The WWW has become a huge repository of information. For almost any knowledge domain there may exist thousands of available sources and billions of data instances. Many of these sources may publish irrelevant data. User-preference approaches have been defined to retrieve relevant data based on similarity, relevance or preference criteria specified by the user. Although many declarative languages can express user-preferences, considering this information during query optimization and evaluation remains as open problem. SQLf, Top-k and Skyline are three extensions of SQL to specify user-preferences. The first two filter irrelevant answers following a score-based paradigm. On the other hand, the latter produces relevant non-dominated answers using an order-based paradigm. The main objective of our work is to propose a unified approach that combines paradigms based on order and score. We propose physical operators for SQLf considering Skyline and Top-k features. Properties of those will be considered during query optimization and evaluation. We describe a Hybrid-Naive operator for producing only answers in the Pareto Curve with best score values. We have conducted initial experimental studies to compare the Hybrid operator, Skyline and SQLf.

1 Introduction

The WWW has motivated the definition of new techniques to access information. Currently, there are around three billion of static documents in the WWW. Some of these documents may publish irrelevant data and users have to be aware to discard the useless information based on their preferences. To express user preference queries many declarative languages have been defined. Those languages can be grouped in two paradigms: score-based and order-based. Score-based languages order the top k answers in terms of a score function that induces a total order. The challenge is to identify the top k objects in this totally ordered set, without having to scan all the objects. On the other hand, order-based languages rank answers using multicriteria selections. Multicriteria induce a partially ordered set or strata; in consequence there is no single optimal answer. Thus the main problem is to construct the first stratum or skyline.

Many algorithms have been proposed to evaluate either score-based or order-based languages, however, some problems still remain open. First, user-preferences may be expressed as combinations of top k and multicriteria selections. To process those queries, a physical operator should identify the top k answers among the objects in the

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 790 – 799, 2005.
© Springer-Verlag Berlin Heidelberg 2005

strata that fulfil the user-preferences, i.e., the proposed operator should unify the functionalities of the score-based and order-based approaches. Second, user preference algorithms need to be integrated into real world query optimizers. Finally, user preferences should be able to evaluate queries against Web data sources. In this paper we propose the definition and implementation of physical operators to achieve these goals.

Given a multicriteria top k query and a set of objects O, we are interested in defining algorithms to efficiently identify the k objects in strata of O. Strata R is a sequence of subsets $<R_1,...,R_n>$, such that R_i is a stratum, $R_i \subseteq O$ and $\cup_{i=1}^{n} R_i = O$, i.e., R is a partition of O. Points in a stratum R_i are non-dominated.

Example 1:
Consider a tourist interested in the top 5 cheap restaurants that are close to his hotel. The tourist can formulate the following top-5 query[1]:

*Select **
From Guide
Skyline of cheap(Price) max, close(Address, HotelAd) max
Order By max(quality(Food))
Stop After 5,

where *cheap* is a user-defined score function that ranks restaurants in terms of their prices. Similarly, *close* is a user-defined function that scores restaurants depending on the distance between a restaurant and the hotel. Finally, *quality* is a user-defined function that measures the quality of the food. Values close to 1 mean that the restaurant is cheap or close or high quality. Note that all the functions are user-dependent. To answer this query, first a query engine should construct the strata of the table *Guide* induced by the multicriteria "cheap(Price) max, close(Address,HotelAd) max". Second, it should construct the first "s" strata $R_1,\ldots,R_s$, such $\left|\cup_{i=1}^{s} R_i\right| \geq 5 \geq \left|\cup_{i=1}^{s-1} R_i\right|$, i.e., the minimum number of strata $R_1,\ldots,R_s$, where the cardinality of their union is greater or equal than 5. Finally, the top 5 answers will be selected from those strata $R_1,\ldots,R_s$. Then, the 5 tuples that maximize the user-defined quality functions will be in the answer. Note that in case of ties, a new score function will be needed to break them.

To the best of our knowledge none of the existing languages express and efficiently evaluate this type of queries. Thus, the main objective of our proposed work, is the definition and integration in a real DBMS (Data Base Management System) of two hybrid operators, a Top-k Skyline Select and a Top-k Skyline Join, that in conjunction with the relational algebra operators will allow users to express and evaluate queries such as Example 1.

In this paper we formalize the problem and present our initial results. The paper comprises 5 sections. In Section 2 we define the problem and provide a naive solution. In Section 3, we briefly describe the existing approaches. In Section 4 we report our initial experimental results. Finally, in Section 0, the concluding remarks and future work are pointed out.

[1] The query is expressed using a combination of the languages defined in [9][18].

2 Motivating Example

Suppose that a research company has two vacancies and has received applications from 5 candidates. Candidates are described by their names, degrees, publications, years of professional experience, grade point averages, and their main research areas. Consider the following relational table that represents this candidate information:

Candidate(Name,Degree,Publications,Experience, GPA, Area)

Additionally, consider the following instances of this relational table with the information of the 5 candidates:

Table 1. Candidates for two vacancies of a research company

Name	Degree	Publications	Experience	GPA	Area
Joseph Lieberman	Post Doctorate	9	2	3.75	Databases
Steve Studer	Post Doctorate	10	1	4	Systems
Margaret Stoffel	PhD.	12	2	3.75	Computer Graphics
Ann Grant	MsC.	13	4	3.6	Networks
Joe Grys	Engineer	6	3	3.25	Databases

According to the company policy, a criterion is not more important than any other, and all of them are equally relevant, hence either a weight or a score function cannot be assigned. A candidate can be chosen for the job if and only if there is no other candidate with a higher degree, number of publications, and years of experience. To nominate a candidate, one must identify the set of all the candidates that are not dominated by any other candidate in terms of these criteria. Thus, tuples in table Candidate must be selected in terms of the values: Degree, Publications, and Experience. For example, Anna Grant dominates Joe Grys because he has worse values in the Degree, Publications and Experience attributes. Thus, the nominates are as follows:

Table 2. Nominate Candidates for two vacancies of a research company

Name	Degree	Publications	Experience	GPA	Area
Joseph Lieberman	Post Doctorate	9	2	3.75	Databases
Steve Studer	Post Doctorate	10	1	4	Systems
Margaret Stoffel	PhD.	12	2	3.75	Computer Graphics
Ann Grant	MsC.	13	4	3.6	Networks

Since the company only has two vacancies, it must apply another criteria to select the two new staff members and discard the other two. Staff members will be selected among nominates in terms of the top two values of one or more overall preference functions that combine values of either the first criteria or the other attributes.

First, considering the maximum GAP as a new preference function, three candidates are the new nominates: Steve Studer, Joseph Lieberman, and Margaret Stoffel. Then, taking into account the degree, the tie between Joseph Lieberman and Margaret Stoffel can be broken, and the selected members will be: Steve Studer and Joseph Lieberman.

Intuitively, to select the staff members, queries based on user preferences have been posted against the table Candidates. There are several databases languages to express preference queries. Skyline, Top-k and SQLf are three user preference languages that could be used to identify some of the staff members. However, none of them will provide the complete set, and post-processing will be needed to identify all the members.

Skyline offers a set of operators to build an approximation of a Pareto curve (strata) or set of points that are not dominated by any other point in the dataset (skyline or first stratum). Thus, by using Skyline, one could just obtain the nominated candidates.

On the hand, SQLf will allow referees to implement a score function and filter some of the winners in terms of the combined function. In order to choose staff members, SQLf computes the score for each tuple without checking dominance relationship between tuples in the dataset. Finally, also Top-k query approaches rank a set of tuples according to some provided functions and do not check dominance relationships. However, it is not possible to define such score function, because all criteria are equally important. Thus, the problem of selecting the staff members corresponds to the problem of identifying the top k elements in partially ordered set.

On one hand, Skyline constructs partially ordered sets induced by the equally important criteria. On the other hand, TopK or SQLf select the best k elements in terms of a score function that induce a totally ordered set. In consequence, to identify the members, a hybrid approach that combines the benefits of Skyline, and SQLf or Top-k is required. Thus, tuples in the answer will be chosen among the stratum induced by a multiple criteria and then, ties will be broken using user-defined functions that eventually induce a total order.

2.1 Research Problem

Our main objective is the definition and integration in a real DBMS of two hybrid operators, a Top-k Skyline Select and a Top-k Skyline Join. We plan to extend a traditional relational cost model with statistics about these operators and make them accessible by a query optimizer.

Given a set $O=\{o_1,\ldots,o_m\}$ of m database objects, where each object o_i is characterized by *p* attributes $(A_1,\ldots,A_p)$; n score-functions $s_1,\ldots,s_n$ defined over some of those attributes, with $s_i : O \rightarrow [0,1]$; a combined score function f defined over subsets of $\{s_1,\ldots,s_n\}$ that induces a total order of the objects in O; and n sorted lists $S_1,\ldots,S_n$ containing all database objects in descending order by score-function s_i respectively, we define *Pareto Points* through recurrence in Definition 1.

Definition 1a (Base Case – First *Pareto Point*):

$$P_1 = \left\{ o_i \in O \,\middle/\, \neg\exists o_j \in O : \begin{pmatrix} s_1(o_i) \le s_1(o_j) \wedge \ldots \wedge s_r(o_i) \le s_r(o_j) \wedge r \le n \\ \wedge \exists q \in [1,\ldots,r] : s_q(o_i) < s_q(o_j) \end{pmatrix} \right\}$$

Definition 1b (Inductive Case: *Pareto Point*):

$$P_i = \left\{ o_l \in O \;\middle/\; \begin{array}{l} o_l \notin P_{i-1} \wedge \neg\exists\ o_t \in \left(O - \cup_{j=1}^{i-1} P_j\right): \\ \left(\begin{array}{l} s_1(o_l) \leq s_1(o_t) \wedge \ldots \wedge s_r(o_l) \leq s_r(o_t) \wedge r \leq n \\ \wedge\ \exists\ q \in [1,\ldots,r]: s_q(o_l) < s_q(o_t) \end{array}\right) \end{array} \right\}$$

We define *Stratum Top-k* in Definition 2 as the minimum number of strata or points of a Pareto Curve that contain the top k answers based on a combined score function f.

Definition 2 (*Stratum Top-k*):

$$StratumTop-k = \left\{ \cup_{i=1}^{j} P_i \;\middle/\; \left|\cup_{i=1}^{j} P_i\right| \geq k > \left|\cup_{i=1}^{j-1} P_i\right| \right\}$$

Finally, the conditions to be satisfied by the answers of a top-k skyline query are given in Definition 3.

Definition 3 (*The Top-k Skyline Problem*):

$$Top-k = \left\{ o_i \in O \;\middle/\; \begin{array}{l} o_i \in StratumTop-k \wedge \\ \neg\exists^{k} o_j \in StratumTop-k: \\ \left(f(o_j) > f(o_i)\right) \end{array} \right\}$$

We have extended the Basic Distributed Skyline Algorithm introduced in [6] in order to construct a set of objects that satisfy the conditions in Definition 3. It is presented in Algorithm 1. Our algorithm first builds all the strata R following Definition 1 and 2, and then, it breaks ties by using function f. Elements are considered with respect to the order induced by R, i.e., the top k answers correspond to the best K objects in the topological sort of R. Topological sorting is done considering the combined score function f.

Algorithm 1**.** (The naive Top-k Skyline Problem)

1. Initialize P_1:=ϕ, n lists $K_1,\ldots,K_n$:=ϕ, and $p_1,\ldots,p_n$:=ϕ.
2. Initialize counters i:=1, nroStrata:=1, s:=1;
   2.1. Get the next object o_{new} by sorted access on list S.
   2.2. If $o_{new} \in P_1$, update its record's i-th real value with $s_i(o_{new})$, else create such a record in P_1.
   2.3. Append o_{new} with $s_i(o_{new})$ to list K_i.
   2.4. Set p_i:=$s_i(o_{new})$ and i:=(i mod n) + 1
   2.5. If all scores $s_j(o_{new})$ $(1 \leq j \leq n)$ are known, proceed with step 3 else with step 2.1.
3. For i=1 to n do
   3.1. While p_i=$s_i(o_{new})$ do sorted on list S_i and handle the retrieved objects like in step 2.2. to 2.3

4. If more than one object is entirely known, compare pairwise and remove the dominated objects from P_1.

   4.1. For j=1 to nroStrata do

      4.1.1 If there are dominated objects in P_j, initialize $P_{j+1}:=\phi$, add dominated objects to P_{j+1} and nroStrata:= nroStrata+1.

5. For i=1 to n do

   5.1. Do all necessary random access for the objects in K_i that are also in all of initialized stratum P_i and discard objects that are not in P_i.

   5.2. Take the objects of K_i and compare them pairwise to the objects in K_i. If an object is dominated by another object remove it from K_i and P_i. Add the dominated object to stratum P_{i+1}.

6. Calculate and order all non-dominated objects by the combined function f in stratum P_s.

7. Output the first K non-dominated objects.

8. While there are not K non-dominated objects, increase s by 1; repeat step 6.

3 Related Work

There exist two paradigms for expressing user preferences: score-based and order-based. Score-based languages rank the top k answers in terms of a score function that induces a total order. The challenge is to identify the top k objects in this totally ordered set, without having to scan all the objects. On the other hand, order-based languages rank answers using multicriteria selections. Multicriteria induce a partially ordered set stratified into subsets of non-dominated objects.

3.1 Order-Based Paradigm

During the 70's and the 80's, people have already studied and proposed user-preference query languages. DEDUCE [17] offers a declarative query language for relational databases including preferences. In [37] DRC (Domain Relational Calculus) is extended with Boolean preference mechanisms and score functions cannot be easily expressed.

Chomicky [20][21] introduced a general logic framework to formalize preference formulas and a preference relational operator called *winnow*. This new operator is integrated into the relational algebra. This approach is more expressive than DRC and implements a combined mechanism between operators. However, the operator *winnow* is not simple for writing and composing preferences. Moreover it is not clear how to implement the operator in a relational system [22].

Preference SQL [33] is an algebraic approach that extends SQL by means of preference operators. Kiessling and Köstler [34] proposed how to extend SQL and XPATH with Preference SQL operators and introduced various types of queries that can be composed with these operators. This language is implemented on the top of a SQL engine. The problem of defining physical operators is not considered.

Skyline operator is introduced in [9] as another SQL extension. This operator expresses preference queries and can be combined with traditional relational operators. Kung et al. defined the first Skyline algorithm in [36], referred to as the maximum vector problem [8][40] and it is based on the divide & conquer principle. Skyline was formalized using a partial order semantic and there are efficient algorithms for relational databases [9][23][25][26][29][35]. Although, the problem of computing the first stratum or skyline is solved, all these algorithms have high time complexity.

3.2 Score-Based Paradigm

Agraval et. al explored how to combine numeric and categorical attributes in [2] and, Agraval and Wimmers [1] introduced a framework for combining preferences. In [41], the preferences are expressions stored as data and evaluated during query execution.

[31] showed how to evaluate preference queries using materialized views that must be defined off-line. This approach does not scale if users define queries and score functions dynamically.

In [4][5][27][28] algorithms are introduced to answer ranking queries. These algorithms assume that inputs are sorted and do not support sources that provide a random access of the data, although these are on common the WWW. Some algorithms for evaluating top-k queries over relational databases are proposed in [3][15]. In [32] a top-k query optimization framework that fully integrates rank-join operators into relational engines is introduced. The algorithms Upper [16], MPro [18] and Framework NC [19] do support random access but are not able to check dominance relationships.

On the other hand, Fagin et al. studied those utility functions that are better with respect to minimization of discrepancy between partial order and weak orders of preferences [28]. In [17], membership functions are defined for measuring tuple adherence to preference conditions. Motro uses functions that measure distance between tuples to measure adherence to goals expressed in the query [38].

Bosc and Pivert integrate the fuzzy set theory with relational algebra. Fuzzy conditions indicate membership grade to the preferences [11]. Later, some query processing mechanisms were proposed [10][12][13][14]; the most relevant is the Derivation Principle due to its lower evaluation cost.

Finally, works that propose to integrate these two paradigms are [7][30]. However, they does not consider the identification of the K best objects in the strata.

4 Initial Results

So far, we have explored the integration of SQLf and Skyline approaches to implement multicriteria top k queries. In this hybrid approach, answers are filtered by means of SQLf, and then a Skyline algorithm is executed. The initial filtering reduces Skyline algorithm complexity time because the Skyline algorithm only has to discard solutions that are not better across all the criteria and that are produced by SQLf. Also, we limit this study only to identify the best objects over the first stratum or skyline.

Our initial experimental study was performed on Oracle 8i. The study consisted of experiments running over one relational table with 100,000 and 1,000,000 tuples. The table has 10 integer columns and one string column. Values of integer columns vary from 1 to 30, where 30 corresponds to the best value. A column may have duplicated values. Duplications are uniformed distributed. The columns are pair-wise statistically independent. We performed 30 randomly generated multicriteria queries. Multicriteria varied between 1, 5 or 9 selections.

Skyline, SQLf and Hybrid were written in PL/SQL and Swi Prolog. Skyline was implemented as the basic SFS algorithm without optimizations [24]. SQLf was evaluated using the Derivation Principle algorithm [12] and implemented on the top of the SQL query engine, and the Hybrid operator was a combination of these two previous algorithms.

The experiments were executed on an Intel 866-MHz PC with 512-MB main memory and an 18-GB disk running Red Hat Linux 8.0.

We report on quality of the answers and query processing time. First, we can observe that the Skyline can return irrelevant objects, while SQLf may either produce irrelevant objects or miss relevant ones. An object is considered irrelevant if it does not belong to the top k skyline objects identified by the Hybrid operator. Between 10% and 30% of the Skyline returned objects were irrelevant. On the other hand, more than 90% of the SQLf objects were irrelevant and less than 10% of the relevant objects were not produced. These initial results motivate the definition of a unified approach that does not produce irrelevant objects or miss relevant ones.

Second, we report on the time taken by Skyline, SQLf and the Hybrid operator to compute the answer. We can observe that the Skyline algorithm time was 3 orders of magnitude greater than the SQLf time, and the Hybrid algorithm time was 2 orders of magnitude less than the Skyline time. These differences may occur because the Skyline algorithm scans the whole data set, while the Hybrid operator scans only the subset of objects produced by SQLf. It has been shown that the best algorithm to compute the full skyline has a (worst-case) complexity of $O(n(\log n)^{d-2})$, where n is the number of points in the data set and d is the number of dimensions [36]. In contrast, the Derivation Principle used to implement the SQLf algorithm just reads a subset of the whole data. The running time of this algorithm depends on the database access time and the score function processing time. The (worst-case) complexity is O(m) where m is the number of points in the answer [12]. Finally, the Hybrid algorithm has a (worst-case) complexity of $O(m(\log m)^{d-2})$. So this task can be very expensive and it is very important to define efficient physical operators.

5 Conclusions and Future Work

In this paper we have described the limitations of the existing approaches to express and evaluate top k multicriteria queries. We have defined our problem and presented a naive solution. We have implemented a first approximation of a unified algorithm as a combination of SQLf and Skyline. To study the performance and the quality of a naive hybrid algorithm, we have conducted an initial experimental study. Our initial results show the quality of the answers identified by the Hybrid operator and the necessity of defining physical operators to efficiently evaluate top k multicriteria queries.

In the future, we plan to define physical Top-k Skyline operators and integrate them into a relational DBMS. Finally, we will extend these operators to access Web data sources.

References

[1] Agrawal, R., and Wimmers, E. L. A framework for expressing and combining preferences. In Proc. of SIGMOD (May 2000), pp. 297-306.
[2] Agrawal, R., Chaudhuri, S., Das, G., and Gionis, A. Automated ranking of database query results. In Proc. of CIDR (Jan 2003).
[3] Aref, W. G., Elmagarmid, A. K., Ilyas, I. F. Supporting Top-k join queries in relational databases. In VLDB Journal (Sep 2004), pp. 207-221.
[4] Balke, W-T., Güntzer, U., and Kiebling, W. Optimizing multi-feature queries for image databases. In VLDB, (Sep 2000), pp. 10-14.
[5] Balke, W-T., Güntzer, U., and Kiebling, W. Towards efficient multi-feature queries in heterogeneous environments. In ITCC (2001), pp. 622-628
[6] Balke, W-T., Güntzer, U., and Xin, J. Efficient Distributed Skylining for Web Information Systems. In EDBT (2004), pp. 256-273.
[7] Balke, W-T. and Güntzer, U. Multi-objetive Query Processing for Database Systems. In Proceedings of VLDB (Sep 2004), pp. 936-947.
[8] Bentley, J. L., Kung, H. T., Schkolnick, M., and Thompson, C. D. On the average number of maxima in a set of vectors and applications. JACM, 25, 4 (1978), pp. 536-543.
[9] Börzönyi, S., Kossman, D., and Stocker, K. The skyline operator. In Proc. of ICDE (Apr. 2001), pp. 421-430.
[10] Bosc, P., and Brisson, A. On the evaluation of some SQLf nested queries. Proceeding International Workshop on Fuzzy Databases and Information Retrieval, 1995.
[11] Bosc, P., and Pivert, O. SQLf: A Relational Database Language for Fuzzy Querying. IEEE Transactions on Fuzzy Systems 3, 1 (Feb 1995).
[12] Bosc, P., and Pivert, O. On the efficiency of the alpha-cut distribution method to evaluate simple fuzzy relational queries. Advances in Fuzzy Systems-Applications and Theory, (1995), 251-260.
[13] Bosc, P., and Pivert, O. SQLf Query Functionality on Top of a Regular Relational Database Management System. Knowledge Management in Fuzzy Databases (2000), 171-190.
[14] Bosc, P., Pivert, O., and Farquhar, K. Integrating Fuzzy Queries into an Existing Database Management System: An Example. International Journal of Intelligent Systems 9, (1994), 475-492.
[15] Bruno, N., Chaudhuri, SL., and Gravano, L Top-k selection queries over relational databases: Mapping strategies and performance evaluation. In TODS 27,2 (2002), pp 153-187.
[16] Bruno, N., Gravano, L, and Marian, A. Evaluating top-k queries over web-accessible databases. In ICDE (2002)
[17] Chang, C. L. Deduce : A deductive query language for relational databases. In Pattern Rec. and Art. Int., C. H. Chen, Ed. Academic Press, New York, 1976, pp. 108-134.
[18] Chang, K. and Hwang, S-W. “Minimal Probing: Supporting Expensive Predicates for Top-k Queries”. Proceedings of the ACM SIGMOD Conference, (Jun 2002).

[19] Chang, K. and Hwang, S-W. "Optimizing access cost for top-k queries over Web sources: A unified cost-based approach". Technical Report UIUCDS-R-2003-2324, University of Illinois at Urbana-Champaign, (Mar 2004).
[20] Chomicky, J. Querying with intrinsic preferences. In Proc. of EDBT (2002), Springer (LNCS 2287), pp. 34-51.
[21] Chomicky, J. Preference formulas in relational queries. ACM TODS 28, 4 (Dec. 2003), 427-466.
[22] Chomicky, J. Semantic optimization of preference queries. In 1st Int. Sym. On Appl. Of Constraint Databases (2004), Springer (LNCS 3074).
[23] Chomicky, J. Godfrey, P., Gryz, J., and Liang, D. Skyline with presorting. In Proc. of ICDE (Mar 2003)., pp. 717-719.
[24] Chomicky, J. Godfrey, P., Gryz, J., and Liang, D. On skyline Computation. (Jun. 2002)
[25] Eng, P. K,. Ooi , B. C., and Tan, K. L. Efficient progressive skyline computation. Proc. Of 27th VLDB (2001), 301-310.
[26] Eng, P. K., Ooi, B. C., and Tan, K. L. Indexing for progressive skyline computation. Data and Knowl. Eng. 46, 2 (2003), pp. 169-201.
[27] Fagin, R. Combining fuzzy information from multiple systems. Journal of Computer and System Sciences (JCSS), 58, 1 (Feb 1996), pp. 216-226.
[28] Fagin, R. Lotem, A., and Naor, M. Optimal aggregation algorithms for middleware. In PODS, Santa Barbara, California (May 2001), pp. 102-113.
[29] Godfrey, P. Skyline cardinality for relational processing. In Proc. of FolKS (Feb 2004), Springer, pp. 78-97.
[30] Goncalves, M and Vidal, M.E. "Preferred Skyline: A hybrid approach between SQLf and Skyline". In Proceedings of DEXA (Ago 2005).
[31] Hristidis, V., Koudas, N., and Papakonstantinou, Y. PREFER: A system for the efficient execution of multi-parametric ranked queries. In Proc. of SIGMOD (May 2001), pp. 259-270.
[32] Ilyas, I.F., Shah, R., Aref, W.G., Vitter, J.S. and Elmagarmid, A.K. Rank-aware Query Optimization. In Proceedings of the 2004 ACM SIGMOD Conf. on Mgmt. of Data, pp. 203 - 214, Paris, France, June 2004.
[33] Kiessling, W. Foundations of preferences in database systems. In Proc. of VLDB (Aug 2002), pp. 311-322.
[34] Kiessling, W., and Köstler, G. Preference SQL: Design, implementation, experiences. In Proc. of VLDB (Aug 2002), pp. 900-1001.
[35] Kossman, D., Ransak, F., and Rost, S. Shooting stars in the sky: An online algorithm for skyline queries. In Proc. of VLDB (Aug 2002), pp. 275-286.
[36] Kung, H. T., Luccio, F., Preparata, F. P. On finding the maxima of a set of vectors. JACM 22, 4 (1975), pp. 469-476.
[37] Lacroix, M., and Lavency, P. Preferences: Putting more knowledge into queries. In Proc. of VLDB (Sept. 1987), pp 217-225.
[38] Motro, A. Supporting goal queries in relational databases. In Proc. of the 1st Int. Conf. on Expert Database Sys. (Apr 1986), pp. 85-96.
[39] Papadimitriou, C. H., and Yannakakis, M. Multiobjective Query Optimization. Proc. ACM SIGMOD/SIGACT Conf. Princ. Of Database Syst. (PODS), Santa Barbara, CA, USA, May 2001.
[40] Preparata, F. P. and Shamos, M. I. Computational Geometry: An Introduction. Springer-Verlag, 1985.
[41] Yalamanchi, A., Srinivasan, J., and Gawlick, D. Managing expressions as data in relational, database systems. In Proc. of CIDR (Jan 2003).

Judicial Support Systems: Ideas for a Privacy Ontology-Based Case Analyzer

Yan Tang and Robert Meersman

Semantics Technology and Applications Research Laboratory (STARLab),
Department of Computer Science,
Vrije Universiteit Brussel,
Pleinlaan 2, B-1050 BRUSSEL 5, Belgium
{ytang1, robert.meersman}@vub.ac.be

Abstract. Nowadays, ontology is applied as an integral part of many applications in several domains, especially in the world of law. The ontology based judicial support system is believed as a useful tool to support, for example, the legal argumentation assistant and legal decision taking in court. The privacy case analyzer is considered as one of the most interesting applications of ontology based privacy judicial support systems. The efficiency of privacy case analyzers depend on several factors such as how to tackle the problem of linking cases to legislations, how to imply the guidance of privacy principles, and how to improve the extraction of cases. This paper addresses those items and describes the research issues that will be investigated challenges of ontology based judicial support systems.

Keywords: Privacy ontology, Woolf reforms, privacy principles, privacy directives, privacy ontology structure.

1 Introduction and Background

Specifications can be beneficial for AI and law in general [12]. As an interesting issue in many technology areas, privacy together with the Internet technology nowadays generates challenging topics such as E-health privacy, E-shopping privacy, transport privacy, Privacy is the ability of an individual or group to prevent information about themselves from becoming known to people other than those they choose to give the information to[1]. According to previous research and workshops of law and coding, the problem of how to bridge the gap between technology and regulation deserves being tackled as listed in [10]. This paper concentrates on the problem of how to apply ontology technology to link privacy cases to legislations.

The remainder of this paper is structured as follows: first, we discuss how a privacy ontology is introduced to represent all these different aspects of common understanding in disparate privacy sub-domains. A description of privacy ontology construction that represents privacy principles, privacy directives, a case analyzer and the relations between them is illustrated together with several challenging research topics. The extensibility of the system with several possible technologies that can be

[1] http://en.wikipedia.org/wiki/Privacy

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 800 – 807, 2005.
© Springer-Verlag Berlin Heidelberg 2005

applied in the system is discussed at the same time. Finally, some research issues, related research fields and plans are outlined.

2 Law and Privacy Ontology Construct

The *privacy ontology* is used to bridge the cases and regulations by capturing knowledge elements (conceptualization) of the privacy domain, intending to improve the quality, transparency, consistency and efficiency of jurisprudence. It covers the semantics of privacy legislations presentation and case reasoning fragments.

Directives are one type of EC legislations. Each state is obliged to give effect to a particular Directive and to transform them into national legislation, usually within a specified period of time [5]. The terminologies used by lawyers are often far away from normal users, although, the gap is narrowing as lawyers and lawmakers are encouraged to use more naturalistic language in *Woolf reforms* [6]. *Privacy directives*, as they are made of 'law language' that has wide social effect than just making lawyers' life more difficult, are needed to have an interpretation mechanism. Privacy ontology plays a role as this interpretation mechanism to bring all those miscellaneous concepts of privacy directives together.

Since law is considered as 'a system of practical reasoning' [5], judges, lawyers and legislators in practice have to deal with a great amount of legal document. Privacy ontology links the real-life cases with the privacy directives and privacy principles to assist legal argument process.

As it is discussed above, the privacy ontology is used 1) to be shared among lawyers in case abstraction system based on shared background vocabulary; 2) to be extended in several privacy-related applications that use the privacy ontology as the operated kernel content. The semantic of privacy ontology and privacy principle meta ontology is used as a guidance of establishing more privacy directives.

Legal reasoning is applied between the layers of facts and the case by the technique of abstraction, where cases are normally divided into several material facts in the legal reasoning process. The *material facts* should be viewed as 'narrow' or 'general', which depends on the *level of abstraction*.

The *privacy principles* are the basis of all privacy (data protection) legislation. The principles have been formulated by the OECD (Organization for Economic Co-operation and Development) in the OECD privacy guidelines of 1980. In 1981 the Council of Europe adopted a privacy protection convention using the same principles. To follow the principles (on which the EU directive 95/46/EC has been based) is the easiest way to apply and build-in privacy legislation into applications, products, and services etc. Applications are built and executed based on those principles. The applications here can be, for example, 1) law or legal information retrieval system; 2) law machine learning system; 3) case reasoning based judicial support system; 4) automatic legal text extraction system; 5) online e-court and online controversy supporting system; 6) law making guidance system; 7) law consult system; etc. All those applications show diverse practices in legislation ontology, accordingly, several challenges appear here such as how to abstract facts from case automatically or semi-automatically; how to link facts to directives and on which abstraction level; how to

use principles as guidance of application execution and directives making; how to prove the quality of the application to see if it really meets the needs; etc.

Fig. 1 shows the privacy ontology structure that is based on the relationship between principles, directives, facts and cases as described in the paragraph above. There are five parts in this privacy ontology structure: privacy applications, legal abstractor, privacy ontology, privacy principle meta-ontology and case parser.

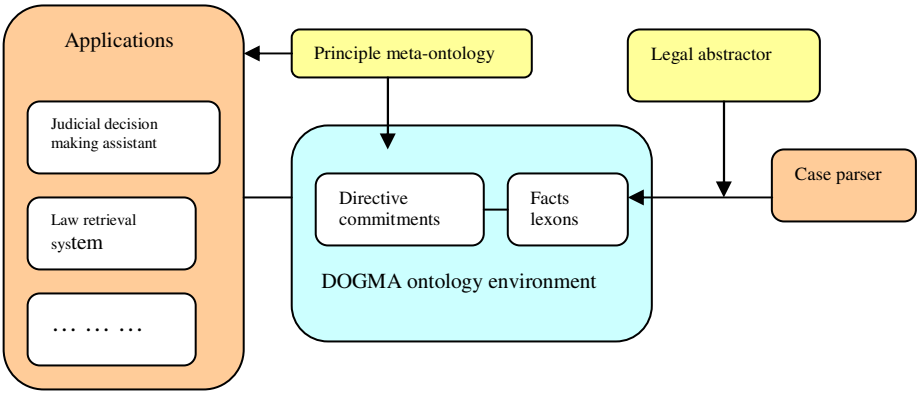


Fig. 1. Privacy ontology structure

The kernel content in this structure is a privacy ontology that adopts the DOGMA framework [4] with separated directive commitments layer and facts lexons layer. The case parser will be an automatic or semi-automatic engine that brings real cases as the inputs and generates coordinated facts. The privacy applications will integrate the privacy ontology via several technologies such as an XML retrieval system. The principle meta-ontology is applied to guide the application design and execution; meanwhile, it can be devoted to guide directive legislators when new directives are made. The legal abstractor lies on the process of case abstraction.

2.1 Privacy Ontology in DOGMA Framework

A privacy ontology is an explicit specification of privacy (deduced from [1], [2]). The role of a privacy ontology is to support privacy case based reasoning. Privacy ontology contains three elements: privacy terms in the taxonomy format, definition of these terms and the privacy axioms. The privacy ontology is still under researching now.

These three elements mentioned in the previous paragraph are represented in DOGMA (Developing Ontology-Guided Mediation for Agents, [3], [4], [8], [9]) environment. DOGMA is a database-driven mechanism of ontology knowledge representation. It considers concepts and relations separately from constraints, derivation rules and procedures.

The DOGMA approach to ontology modeling consists of two layers: the ontology base and the ontological commitments. The ontology base is sets of lexons that repre-

sent the entities of conceptualization. The ontology commitments contain the lexon selection, organization, instantiation and system context.

A lexon is a quintuple $<\gamma, t_1, r_1, r_2, t_2>$, where $\gamma \in \Gamma$, $t_1, t_1 \in T$, $r_1, r_2 \in R$. t_1, t_1 are terms that represent two concepts. r_1, r_2 are roles referring to the relationships that the concepts share with respect to one another. γ is a context identifier in which the concepts identified by terms t_1, t_1 and roles r_1, r_2 become meaningful. Accordingly, Γ is a context set, T is a terms set and R is a roles set.

Privacy ontology hence has two layers: the privacy directive commitment and the privacy facts lexons. The directive commitment layer is based on the shared knowledge of privacy directives together with privacy rules.

Fact Lexons

The fact lexons here are extracted from the case under the guidance of legal abstractor. They are considered as many 'isolated' groups of entities that are represented two by two and can be stored in any database systems.

Directive Commitment

Directive commitment tailors those fact lexons logically ascribing to real life application requirements. For instance, in judicial decision making support system, to establish the existence of certain facts is the first thing that a court has to do. The *fact proving* hence is one of the directive commitment components.

A second component can be the *syntax interpreting*. An isolated word cannot make the lawyers to work in the process of interpretation because of the constituency and complexity of the natural human's linguistic structure. There are a number of difficulties arising, such as *syntactic ambiguity*. Those problems have been addressed by some pioneers, e.g. Bryan Niblett, who explored the problem of syntactic ambiguity by comparing the syntax of a legal fact to that of a computer program [13]. But in practice, the syntactic ambiguity has to be resolved by the courts by making choices on competing interpretations. More problems of syntactic ambiguity may arise as case recorders can grow monotonically, which can be considered as an interesting researching issue in this component.

A third component is *interpretation & justification* in legal argument. This component regards the rules in a case and the facts to which they apply. Those case rules should be established in a delegate way for further usage in some applications, e.g. the legal argumentation assistant system. Here, several interesting research issues come up: how to present those rules in machine understandable and human understandable codes. As the legal argument needs grounding reasons which provide a justification for the position to be taken, so the interesting research issue can be: how to realize those reason grounding mechanisms, what if the *consequential* form or *rightness reasons* form [7] is taken.

A very abstract and useful component in those four components might be the *fact reasoning*. The fact reasoning in court is an ambiguous process in the sense that lot of elements that influence the witness testimony; nevertheless, this problem is simplified by the process of data protection in the privacy domain with several useful technologies such as a tracing system and an authentication system. In that case, the existence of facts is proven by third party (the controller); hence, it's easier to address various problems of uncertainty and reliability on the facts.

The *fact proving* and the *syntax interpreting* are used in the fact-finding process. The *interpretation & justification* and the *fact reasoning* are more abstract and theoretical. We only show four models in the directive commitment module; however, this commitment layer is not restricted to those four components. They are extended according to the real life applications while the directive commitment layer can be aggrandized to the legislation commitment, then it becomes more complicated and rich.

2.2 Legal Abstractor

The legal abstractor is defined as the extraction system of facts at three essential level of abstraction. It bridges the case parser and privacy ontology by formalizing the relationships between concepts of cases and legal ontology.

The extraction of case into facts needs a process of language so that the linguistic approach should be taken into consideration on this layer, thus, 1) a challenge of extraction the miscellaneously presented case arises. In privacy domain, as the cases might be described by the data controller or the third party, the standards of explaining cases should be established. Accordingly, 2) how to make such a kind of privacy case describing standards is another topic that deserves researching.

2.3 Principle Meta-ontology

The principle meta-ontology layer is defined as the guidance of privacy directives making and privacy applications design. The UML use case diagram is adopted to show the conceptual modeling of a principle meta-ontology (see Fig. 2). The privacy controller is supposed to control the application design and execution, the user information and data tracing and the law making process.

It should be noted that we use meta-ontology instead of formal ontology. The formal ontology ([1], [2]) has precise formal semantics and logical meanings, and,

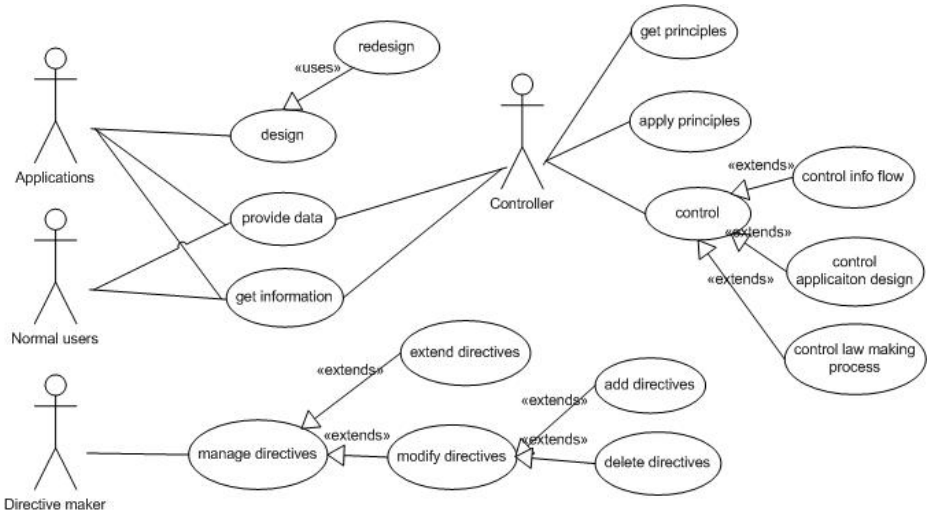


Fig. 2. Principle use cases model

DOGMA is a framework of formal ontology presentation [4]. Meta-ontology is neither *data concept* nor *natural concept*. It thus comes up a researching topic that how to present the meta-ontology as it itself is the ontology of formal ontology.

From Fig. 2, it's quite clear that the controller is a crucial actor in the whole model, thus the principle meta-ontology should focus on the functionality and concepts of the controller.

As a meta-ontology, the same interesting research issue arises as the section 2.2 explains, 1) that is: how to represent the meta-ontology? Which framework is the most suitable one for meta-ontology? Moreover, if domain extension is deliberated, 2) how can we apply different principles in different legislations? The principles of data protection directives maybe very near the privacy principles discussed here, but what will happen if the structure is expanded to human right principles, which is the base of every law?

2.4 Application Design

As section 2.3 describes, the privacy application design should be under the control of principles. The data structure layout, the flow charts, the entity relationship diagram of coding those layouts, etc. are counted in the application design process. Those design tools and methods show the requirements of applications, which means you should know what the applications look like before starting designing them. Below shows some typical privacy application design listed.

Law Retrieval System
A law retrieval system is split into several sub retrieval systems such as a privacy directive retrieval system. The privacy retrieval system is heavily based on the database design of the fact lexons and on the directives commitment design. Thus, the engineering proposal can be how to design the privacy retrieval system under the guidance of principle meta-ontology, which database schema is used to represent fact lexon and how principle meta-ontology is represented etc.

E-Court
E-court is seen as a system used to retrieve documents from government or a virtual court debate system. E-court is still new and promising today so that it brings an issue: "Can we imitate a real court?" The virtual court can be the Court of Appeal, the Court of First Instance, County Court, Crow Court, European Court of Human Rights, High Court, Magistrate's Court, Privy Council and European Court of Justice etc. Accordingly, more challenges show up as: 1) how can we define functions of the virtual court? 2) How can the system emulate the precedents of the court?

2.5 Case Parser

The case parser is a user interface to the privacy ontology base. Legal abstractor functions as the technology of 'interpretation' between case and facts. In that case, the case parser is considered as an application that realizes the interpretation of the legal abstractor. As we have discussed in section 2.3, the presentation of the legal abstractor is an interesting researching issue.

3 Discussion

As we have already discussed in section 2.1, the privacy ontology will be presented in the framework of DOGMA. A privacy ontology extraction methodology will be developed. The facts lexons will be stored in the DOGMA Studio[2] and all the components of a directive commitment will be represented in ORM (Object Role Modeling) schema [11], UML (Unified Modeling Langue) and XML schema. Several rules and reasoning mechanisms will be applied in that directive commitment in a visualized way.

The application of law retrieval system shown in section 2.4 will be made later on. Nowadays, there are some law retrieval systems available, e.g. the commonwealth law retrieval system in Australia[3]. We are going to make a privacy retrieval system that takes a case as an input and generates a hit directive list. At the same time, an ontology application design method will be made.

4 Conclusion and Future Work

In this paper, we have discussed a privacy ontology based case analyzer system in a privacy ontology structure. The privacy case analyzer is one of the ontology based judicial supporting applications. It bridges the gap between facts and law in an automated way.

The privacy ontology structure as Fig. 2 shows can be extended for the usage of legislation ontology when this kind of legislation has principles guidance for law making and application design. The privacy case analyzer is built based on this structure. We have discussed the privacy ontology structure in detail. Several interesting issues and research fields are brought up during the explanation of the privacy ontology structure.

The legal abstractor (section 2.2) and the case parser (section 2.5) are considered together as an expert system that will be developed in the future. Both of them are considered as the parts of the privacy case analyzer. The legal abstractor takes cases from the case parser under some abstraction rules in a natural language processing and generates a certain number of lexon tables. The abstraction rules will be studied and explored in the linguistic domain. The case parser takes cases from users and puts them in the case depository, so the case parser user interface and case depository database will be designed.

I am in the first academic year of PhD and the future work of my PhD research will focus on 1) an ontology application design method, 2) presentation of principle meta-ontology, 3) privacy principle guidance rules making and standardizing, 4) legal abstractor and case extractor development, 5) privacy retrieval system development. Details about the research domain have been presented in section 2.

Acknowledgements. It's our pleasure to thank Gang Zhao, Aldo De Moor and John Borking for their comments on ideas represented in this paper. The authors are

[2] Still under development.
[3] http://www.comlaw.gov.au/

grateful to Peter Spyns for ontology discussion and paper editing. A very special thank is to Jean Lam for his encouragement for life. This research is partly supported by the EU FP6 IP Prime (IST-2002-507 591) with the EU JRC subcontract (22407-2004-10 F1SC ISP BE).

References

1. T. R. Gruber. *A translation approach to portable ontologies*. Knowledge Acquisition, 5(2):199-220, 1993.
2. T. R. Gruber. *Toward principles for the design of ontologies used for knowledge sharing*. Workshop on Formal Ontology, Padova, Italy, 1992.
3. P. De Leenheer & A. De Moor, *Context-driven Disambiguation in Ontology Elicitation*. In, P. Shvaiko & J. Euzenat,(eds.), *Context and Ontologies: Theory, Practice and Applications*, AAAI Technical Report WS-05-01, pp. 17 - 24, AAAI Press, 2005.
4. R. Meersman, *Ontologies and Databases: More than a Fleeting Resemblance*. In, A. D'Atri & M. Missikoff,(eds.), OES/SEO 2001 Rome Workshop, Luiss Pub., 2001.
5. J. A. Holland, J. S. Webb, *Learning legal rules*, Oxford university press, 2003
6. L. Woolf, *substantial change of civil court rule in UK*, 1999.
7. R. Summers, *Two types of substantive reasons: the core of a theory of common law justification*, 63 Cornell Law Review 707, 1978.
8. P. Verheyden, J. De Bo & R. Meersman, *Semantically unlocking database content through ontology-based mediation* . In, C. Bussler, V. Tannen & I. Fundulaki,(eds.), Semantic Web and Databases: 2nd Int'l Workshop, SWDB 2004, Toronto ,Canada, 2004, Revised Selected Papers, LNCS 3372, pp. 109 - 126, Springer Verlag, 2005.
9. P. De Leenheer & R. Meersman, *Towards a formal foundation of DOGMA Ontology Part I: Lexon Base and Concept Definition Server*. Technical Report STAR-2005-06, STAR-Lab, Brussel, 2005.
10. J. Dumortier, C. Goemans, *Roadmap for European legal research in privacy and identity management*, K.U. Leuven, 2003.
11. T. A. Halpin. *Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design*, San Francisco, California, Morgan Kaufman Publishers, 2001.
12. A. Valente, J. Breuker, *Ontology: the missing link between legal theory and AI & law*, in A. Soeteman (eds.), Legal knowledge based systems JURIX 94: Lelystad, Koninklijke Vermande, 1994.
13. B. Niblett, *computer science and law: an introductory discussion*, in B. Niblett (ed.), computer science and law: an advanced course, Cambridge: Cambridge University Press, 1980.

SWWS 2005 PC Co-chairs' Message

Welcome to the Proceedings of the first IFIP WG 2.12 & WG 12.4 International Workshop on Web Semantics (SWWS'05). This book reflects the issues raised and presented during the SWWS workshop which proves to be an interdisciplinary forum for subject matters involving the theory and practice of web semantics. A special session on Regulatory Ontologies has been organized to allow researcher of different backgrounds (such as Law, Business, Ontologies, artificial intelligence, philosophy, and lexical semantics) to meet and exchange ideas.

This first year, a total of 35 papers were submitted to SWWS. Each submission was reviewed by at least two experts. The papers were judged according to their originality, validity, significance to theory and practice, readability and organization, and relevancy to the workshop topics and beyond. This resulted in the selection of 18 papers for presentation at the workshop and publication in this proceedings. We feel that these Proceedings will inspire further research and create an intense following. The Program Committee comprised: Aldo Gangemi, Amit Sheth, Androod Nikvesh, Mihaela Ulieru, Mohand-Said Hacid, Mukesh Mohania, Mustafa Jarrar, Nicola Guarino, Peter Spyns, Pieree Yves Schobbens, Qing Li, Radboud Winkels, Ramasamy Uthurusamy, Richard Benjamins, Rita Temmerman, Robert Meersman, Robert Tolksdorf, Said Tabet, Stefan Decker, Susan Urban, Tharam Dillon, Trevor Bench-Capon, Usuama Fayed, Valentina Tamma, Wil van der Aalst, York Sure, and Zahir Tari.

We would like to express our deepest appreciation to the authors of the submitted papers and thank all the workshop attendees and the program committee members for their dedication and assistance in creating our program and turning the workshop into a success. Producing this book would not have been possible without the much appreciated contribution of Kwong Lai.

Thank you and we hope you will enjoy the papers as much as we do.

August 2005 Tharam Dillon (IFIP WG 2.12 Chair), University of Technology Sydney
Ling Feng (Co-Chair), University of Twente
Mustafa Jarrar, Aldo Gangemi, Joost Breuker, Jos Lehmann, André Valente (Vice-Chairs)
(SWWS+WORM Committee)

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, p. 808, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Adding a Peer-to-Peer Trust Layer to Metadata Generators

Paolo Ceravolo, Ernesto Damiani, and Marco Viviani

Università di Milano, Dipartimento di Tecnologie dell'Informazione,
Via Bramante, 65, 26013 Crema, Italy
{ceravolo, damiani, viviani}@dti.unimi.it

Abstract. In this paper we outline the architecture of a peer-to-peer Trust Layer that can be superimposed to metadata generators producing classifications, like our ClassBuilder and BTExact's iPHI tools. Different techniques for aggregating trust values are also discussed. Our ongoing experimentation is aimed at validating the role of a Trust Layer as a non-intrusive, peer-to-peer technique for improving quality of automatically generated metadata.

Keywords: Knowledge Extraction, Metadata Validation, Trust, P2P.

1 Introduction

Today's growing interest in studying and developing systems for generating and managing metadata is strictly connected to the need of share and manage knowledge about data. Typically, metadata provide annotations specifying content, quality, type, creation, and spatial information of a data item. Though a number of specialized formats based on *Resource Description Framework* (RDF) are available, metadata can be stored in any format such as free text, *Extensible Markup Language* (XML), or database entries. There are a number of well-known advantages in using information extracted from data instead of data themselves. On one hand, because of their small size compared to the data they describe, metadata are more easily shareable than data. Thanks to metadata sharing, information about data becomes readily available to anyone seeking it. Thus, metadata make data discovery easier and reduces data duplication. On the other hand, metadata can be created by a number of sources (the data owner, other users, automatic tools) and may or may not be digitally signed by their author. Therefore, metadata have non-uniform trustworthiness. In order to take full advantage of metadata it is fundamental that (i) users are aware of each metadata level of trustworthiness (ii) metadata trustworthiness is continuously updated, e.g. based on the view of the user community. This is even more important when the original source of metadata is an automatic metadata generator whose error rate is in general not negligible like our *ClassBuilder*[12], developed in the framework of the KIWI project, or BTExact's *iPHI*[2]. The aim of the *ClassBuilder* platform is intelligent clustering of an information flow. The clustering process

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 809–815, 2005.
© Springer-Verlag Berlin Heidelberg 2005

compares documents according to the structure as well to the content. Content-based clustering relies on a hybrid technique between traditional *k-mean* and *nearest-neighbor* algorithms. Structure-based clustering employs advanced fuzzy logic techniques, described in detail in [1]. Once clusters have been computed, each cluster-head is used to generate the assertions that compose the description of a typical instance of the domain.
The present paper briefly outlines our current research work (for a more detailed description, see [13]) on how to validate such assertions (as well as any other automatically generated classification metadata) by means of a *Trust Layer*, including a *Trust Manager* able to collect human behavior in the navigation of data (or metadata) and to compute variations to trust values on metadata. In our approach, monitoring of user behavior is complemented with explicit peer-to-peer voting on metadata. Depending on the desired level of anonymity, votes may be uniform or based on user roles. As far as experimentation is concerned, the Trust Layer can be validated either by programmable agents playing the roles of users navigating data and metadata or by a trust display module, called Publication Center, capable of showing dynamic trust landscapes on metadata and of computing trust-based views on them. The paper is organized as follows: in Section 2 we outline the architecture of our Trust Layer, while Section 3 addresses the problem of aggregating trust values expressed by different peers at various levels of anonymity. Finally, in Section 4 we draw the conclusion and point to our ongoing research work on P2P trust-based systems.

2 The Trust Layer architecture

Before describing our proposed Trust Layer, let us make some short remarks on related work. Current approaches distinguish between two main types of trust management systems [5], namely *Centralized Reputation Systems* and *Distributed Reputation Systems*. In centralized reputation systems, trust information is collected from members in the community in the form of ratings on resources. The central authority collects all the ratings and derives a score for each resource. In a distributed reputation system there is no central location for submitting ratings and obtaining resources' reputation scores; instead, there are distributed stores where ratings can be submitted. In a "pure" peer-to-peer setting, each user has its own repository of trust values on the resources she knows. Our peer-to-peer approach is different only inasmuch trust is attached to metadata in the form of assertions rather than to generic resources. While trust values are expressed by peers, our Trust Layer includes a centralized Metadata Publication Center that acts as an index, collecting and displaying metadata assertions, possibly in different formats and coming from different sources. It is possible to assign different trust values to assertions depending on their origin: assertions manually provided by a domain expert are more reliable than automatically generated ones. Metadata in the Publication Center are indexed and peers interact with them by navigating them and providing implicitly (with their behavior) or explicitly (by means of an explicit vote) an evaluation about metadata trustworthiness. This

trust-related information is provided by the Publication Center to the Trust Manager in the form of new assertions. Trust assertions, which we call *Trust Metadata*, are built using the well-known technique of *reification*. This choice allows our system to interact with heterogeneous sources of metadata: our Trust Metadata are not dependent on the format of the original assertions. Also, all software modules in our architecture can evolve separately; taken together, they compose a complete Trust Layer, whose components communicate by means of web services interfaces. This makes it possible to test the whole system despite the fact that single models can evolve with different speeds. Summarizing our architecture, the Trust Manager is composed of two functional modules:

- *Trust Evaluator*: examines metadata and evaluates their reliability;
- *Trust Aggregator*: aggregates all the inputs coming from the trust evaluator peers by means of a suitable aggregation function.

Our Trust Manager is the computing engine behind the Publication Center that provides an overview on metadata reliability.

2.1 Interacting with the Trust Layer

We are now ready to outline the main phases of an interaction with our Trust Layer.

- *Metadata Selection*: In the first phase, peers navigate a collection of heterogeneous metadata selecting assertion (or assertion patterns) they find interesting. Whatever their format, assertions say something about resources; therefore, the result of this phase is a non-filtered set of resources indexed by the selected assertion(s).
- *Metadata Rating*: According to their opinion about results, peers may explicitly (e.g. by clicking on a `Confirm association` button) assign a rating to the quality of the assertion related to the query. More frequently, users will simply click on data they are interested in in order to display them; we take this action as an implicit vote confirming the quality of the association between data and metadata (after all, it was the metadata that allowed the user to reach the data). Of course, explicit votes should count more than implicit ones.
- *Rating Aggregation*: aggregation should be performed taking into account peers' profiles (when available) and other context metadata.

We shall elaborate on these issues in the next Section. For now, it suffices to say that once enough votes have been collected and aggregated[1], the Trust Manager is able to generate Trust Metadata assertions and associate global trust values to them. Each Trust Metadata assertion specifies the trust value attached to

[1] Of course, secure voting protocols are needed to prevent tampering with votes in transit and other security attacks. Security issues are outside the scope of this paper; the general structure of a secure voting protocol for P2P environments can be found in [6].

an original assertion, the latter being now reified as the object of the Trust Metadata assertion. After this phase, since trust values are now available, peers can interact with the Trust Manager and compute *Trust Constraints*, i.e. views on the original metadata based on our Trust Metadata (e.g. all metadata whose trust value TV satisfies $TV \leq x$). An aggregation function is used to collect votes and update trust values during the whole system life; hence trust constraints are continually updated as well.

3 The Reputation Computation Problem

There are two main problems in managing trust information about metadata in a peer-to-peer environment: how to collect trust values and how to aggregate them. As we discussed in the previous section, our system collects ratings based on implicit and/or explicit behavior of users that approach the Publication Center. Ratings deriving from implicit user behavior can be computed for example by the time spent by the user working on a resource. On the other hand, explicit votes are usually collected from a subset of the users, depending on their role or their knowledge about the domain (or both of them). Anyway, the main problem to solve, once trust values have been obtained, is how to aggregate them. First of all it is necessary to consider if the system works with anonymous users or not. In the former case, every user behavior contributes in the same way to calculate a unique trust value on a resource. In the latter case ratings have to be aggregated initially at user level, and subsequently at global level. Several principles for computing reputation and trust measures have been proposed [5]:

- *Summation or Average of ratings.* It is simply the sum of positive and negative ratings separately, keeping a total score as the positive score minus the negative score.
- *Bayesian Systems.* This kind of systems take binary ratings as input and are based on computing reputation scores by statistical updating of beta probability density functions (PDF).
- *Discrete Trust Models.* They are based on human discrete verbal statements to rate performances (ex. *Very Trustworthy*, *Trustworthy*, *Untrustworthy*, *Very Untrustworthy*).
- *Belief Models.* Belief theory is a framework related to probability theory, but where the sum of probabilities over all possible outcomes non necessarily add up to 1, and the remaining probability is interpreted as uncertainty.
- *Fuzzy Models.* Linguistically fuzzy concepts can represent trust, where membership functions describe to what degree an agent can be describes as, for example, trustworthy or not trustworthy.
- *Flow Models.* They compute trust by transitive iteration through looped or arbitrarily log chains.

In our system we assume that the level of trust for an assertion is the result of the aggregation of *fuzzy values* representing human interactions with the original metadata. Choosing the aggregation operator, however, is by no means straightforward. Arithmetic average perform a rough compensation between high and

low values; when numerical values have to be combined, in particular in the case of fuzzy membership values (as in our case), the basic operations which are involved are the *Weighted Mean* [3] and the *Ordered Weighted Averaging operator* [10], analyzed in [8][7]. The difference between the two functions is in the meaning they assign to the weights that have to be combined with the input values. The weighted means allows the system to compute and aggregate value from the ones coming from several sources, taking into account the reliability of each information source. The OWA operator is a weighted average that acts on an ordered list of arguments and applies a set of weights to tune their impact on the final result: it allows the user to weight the values supplied in relation to their relative ordering[2]. In our case, we start with a set of ratings $\{r_{t_i}\}$, where t_i is the rating timestamp, and get:

$$f_{\mathrm{OWA}} = \frac{\sum_{k=1}^{n} w_k r_{t_k}}{\sum_{k=1}^{n} w_k} \tag{1}$$

In the above equation, n is cardinality of the reputation set (i.e., the number of reputations to be aggregated) and reputations are listed in decreasing order ($r_{t_1} \leq r_{t_2}, ..., \leq r_{t_n}$). Finally, $w = [w_1, w_2, ...w_n]$ is a weighting vector. In our recent work, OWA-based aggregation for ratings has been successfully compared with other techniques, including probabilistic ones[4]. Unfortunately, it can be applied only in a fully anonymous P2P setting, i.e., one where all the users are equal. In fact, in the OWA, weights measure the importance of a value (in relation to other values) based on its size, i.e. disregarding the information source that has expressed it. In our system there is the possibility that different types of peers express votes on a resource and, therefore, it is important to weight votes according to each peer's reliability (computed for example based on its role). Moreover votes can be aggregated differently depending on a number of other criteria (e.g. peer location, connection medium, etc.). In this case, the *Weighted OWA operator* (WOWA) [9] seems to be a better solution, because it combines the advantages of the OWA operator and the ones of the weighted mean. WOWA uses two sets of weights: $\mathbf{p}$ corresponding to the *relevance of the sources*, and $\mathbf{w}$ corresponding to the *relevance of the values*.

Definition 1. *Let* $\mathbf{p}$ *and* $\mathbf{w}$ *be weighting vectors of dimension* n *(*$\mathbf{p} = [p_1\ p_2 ... \ p_n]$, $\mathbf{w} = [w_1\ w_2\ ...\ w_n]$*) such that (i)* $p_i \in [0,1]$ *and* $\sum_i p_i = 1$*; (ii)* $w_i \in [0,1]$ *and* $\sum_i w_i = 1$.

In this case a mapping $f_{\mathrm{WOWA}} : \mathbb{R}^n \rightarrow \mathbb{R}$ *is a Weighted Ordered Weighted Averaging (WOWA) operator of dimension* n *if*

$$f_{\mathrm{WOWA}}(a_1, a_2, ..., a_n) = \sum_i \omega_i a_{\sigma(i)} \tag{2}$$

[2] For example, as described in [11], it is possible to obtain an average much like the score assigned by judges in certain sports in the Olympic Games, i.e. one that cuts out extreme values.

where $\{\sigma(1), \sigma(2), ..., \sigma(n)\}$ *is a permutation of* $\{1, 2, ..., n\}$ *such that* $a_{\sigma(i-1)} \geq a_{\sigma(i)}$ *for all* $i = 2, ..., n$.
With $a_{\sigma(i)}$ *we indicate the* i^{th} *largest element in the collection* $\{a_1, a_2, ..., a_n\}$ *and the weight* ω_i *is defined as*

$$\omega_i = w^* \left(\sum_{j \leq i} p_{\sigma(j)} \right) - w^* \left(\sum_{j < i} p_{\sigma(j)} \right) \tag{3}$$

with w^* *a monotone increasing function (e.g. a polynomial) that interpolates the points* $(i/n, \sum_{j \leq i} w_j)$ *together with the point* $(0, 0)$. ω *is used to represent the set of weights* $\{\omega_i\}$: $\omega = \{\omega_1, \omega_2, ..., \omega_n\}$.

Compared with simple OWA, the WOWA operator allows to model more complex situations, e.g. ones in which not all voters are equal, regardless of the vote they express. Also, it will allow to model the Trust Manager "attitude" (i.e., more or less confident) with respect to users expressing votes[3].

Following Definition 1, it is necessary, given a set or weights $\mathbf{p}$ and $\mathbf{w}$, and the data vector $\mathbf{a}$, to define $S = \{(i/n, \sum_{j \leq i} w_j) | i = 1, ..., n\} \cup \{0, 0\}$ and the function w^* interpolating S. There are two possible ways to proceed: (i) to establish the weighting vector $\mathbf{w}$ and then to interpolate w^* or (ii) to begin with the definition of w^*. In the first case, to interpolate the weighting points, it is adequate any method that from monotone data and bounded in the unit interval defines a monotone and bounded function. In the second approach it is possible to derive the set of weights ω from w^*, where w^* is any monotone increasing function within the $[0, 1]$ interval with $w^*(0) = 0$ and $w^*(1) = 1$.

Referring to the first method, having for example a set of input values $\mathbf{a} = [a_1\ a_2\ a_3\ a_4]$ already ordered from the maximum to the minimum (avoiding in this way the ordering process), it is possible to choose the weighting vector $\mathbf{w}$ depending on the importance we want assign to data values. Choosing for example a vector $\mathbf{w}' = [.1\ .4\ .4\ .1]$ we consider more important central values than extreme ones. On the contrary $\mathbf{w}'' = [.4\ .1\ .1\ .4]$ indicates that extreme values are the most important. The WOWA operator allows flexible modeling of different levels of expertise on the part of the peers expressing their votes.

4 Conclusion

In this paper, we presented our approach for developing a Trust Layer for improving the quality of automatically generated metadata based on user feedback. Different aggregation operators were discussed. Extensive experimentation of the approach described in this paper is currently ongoing; preliminary results are encouraging.

[3] This attitude, together with other factors, might well be expressed by an additional layer of service metadata describing the Trust Manager operation. This would allow for searching and selecting a Trust Manager service based on a summary of its aggregation technique.

Acknowledgments

This work was partly funded by the Italian Ministry of Research Fund for Basic Research (FIRB) under project RBAU01CLNB_001 "Knowledge Management for the Web Infrastructure" (KIWI). We thank the project partners for their valuable input. Thanks are also due to our BTExact colleagues Ben Azvine, Trevor Martin and Zahn Cui for many interesting discussions on knowledge management issues.

References

1. P. Ceravolo, M.C. Nocerino and M. Viviani : Knowledge extraction from semi-structured data based on fuzzy techniques. Eighth International Conference on Knowledge-Based Intelligent Engineering Systems (KES 2004), Wellington, New Zealand, (2004) 328–334.
2. T. P. Martin and B. Azvine, Acquisition of Soft Taxonomies for Intelligent Personal Hierarchies and the Soft Semantic Web, BT Technology Journal, Volume 21, Number 4, December 2003 pp. 113 - 122
3. J. Aczél : On Weighted synthesis of judgements. Aequationes Math. **27** (1984) 288–307
4. R. Aringhieri,E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati: Fuzzy Techniques for Trust and Reputation Management in Anonymous Peer-to-Peer Systems. JASIST (to appear) (2006)
5. Audun, J., Roslan, I., Boyd, C.: A Survey of Trust and Reputation Systems for Online Service Provision. Decision Support Systems (to appear) (2005)
6. E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati: Managing and Sharing Servents' Reputations in P2P Systems. IEEE Trans. Knowl. Data Eng. **15(4)** (2003) 840–854
7. J. Fodor, J. L. Marichal, M. Roubens : Characterization of the Ordered Weighted Averaging Operators. IEEE Trans. on Fuzzy Systems **3(2)** (1995) 236–240
8. M. Grabisch: Fuzzy integral in multicriteria decision making. Fuzzy Sets and Systems **69** (1995) 279–298
9. V. Torra: The weighted owa operator. International Journal of Intelligent Systems **12(2)** (1997) 153–166
10. R. Yager: On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE Transactions on Systems, Man and Cybernetics **18(1)** (1988) 183–190
11. R.Yager: Quantifier Guided Aggregation Using OWA operators. International Journal of Intelligent Systems **11**
12. P. Ceravolo, A. Corallo, E. Damiani, G. Elia, M. Viviani, A. Zilli: Bottom-up extraction and maintenance of ontology-based metadata, In Capturing Intelligence: Fuzzy Logic and the Semantic Web, Elie Sanchez, ed., Elsevier.
13. P. Ceravolo, E. Damiani, M. Viviani: Adding a Trust Layer to Semantic Web Metadata To appear in Soft Computing for Information Retrieval on the Web, F. Crestani, E. Herrera-Viedma, G. Pasi, eds., Elsevier.

Building a Fuzzy Trust Network in Unsupervised Multi-agent Environments

Stefan Schmidt[1], Robert Steele[1], Tharam Dillon[1], and Elizabeth Chang[2]

[1] University of Technology, Sydney, PO Box 123 Broadway, NSW 2007 Australia
{sschmidt, rsteele, tharam}@it.uts.edu.au
[2] Curtin University of Technology, Perth, GPO Box U1987, WA 6845 Australia
e.chang@curtin.edu.au

Abstract. In automated and unsupervised multi-agent environments, where agents act on behalf of their stakeholders, the measurement and computation of trust is a key building block upon which all business interaction scenarios rely. In environments, where the individual and independent calculation of trustworthiness values for future negotiation partners is desired, flexible algorithms and models imitating human reasoning are crucial. This paper introduces a trust evaluation model that imitates human reasoning by using fuzzy logic concepts. Furthermore, post-interaction processes such as business interaction reviews and credibility adjustment are used to continuously build and refine an information repository for future trust evaluation processes. Fuzzy logic offers a mathematical approach encompassing uncertainty and tolerance of imprecise data, and combined with our highly customizable model, it allows to meet the security needs of different stakeholders.

1 Introduction

Intelligent agents are designed to act on behalf of users in multi-agent systems such as e-commerce markets; therefore, they need to mimic various abilities of the human mind. One characteristic of computational logic, however, represents its precision and certainty, which is contrary to the reasoning based on approximations and uncertainties applied by human beings. Hence, the interaction process between two agents should imitate the conventions of human behavior and thus needs to emulate reasoning based on tolerance and approximations. Fuzzy logic represents a promising concept to close the gap between human reasoning and computational logic [3].

A scenario, where an agent is given the task of purchasing a specific book in an e-commerce marketplace, would be a typical application for our proposed model. After querying registries and consulting with other agents, it might have found five services which offer the sought-after product with similar prices and delivery conditions. This leaves the agent to decide with which service it wants to enter into a direct business interaction. This business interaction represents several formal commitments including the negotiation of a legally binding contract and the processing of payments. Following our proposed model, the agent performs a trust evaluation procedure for each of the discovered services. Using these results, the agent decides with which service it enters into the actual business interaction.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 816 – 825, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Variables like trustworthiness, credibility, and confidence imply subjectivity as well as uncertainty and they cannot be measured as crisp values; however, their calculation is still highly desirable. In the context of trust measurement for autonomous agent interactions one is interested in processing this imprecise input to calculate the degree of trust in surrounding agents [1, 5]. Fuzzy logic represents a mathematical approach to deal with uncertainty in the decision-making process by using imprecise numbers (fuzzy numbers) to express the membership to a context [2].

Currently, research and analysis is undertaken in the area of automated e-commerce negotiation models, but problems concerning trust evaluation for unsupervised interactions are largely disregarded. This paper will provide a model for the computation of trustworthiness in multi-agent systems using fuzzy concepts. In addition, we integrate post-interaction processes like business interaction reviews and credibility adjustment which are designed to continuously build and refine an information repository for future trust evaluation processes.

2 Background

The measurement of trust is currently not satisfactorily reflected in multi-agent environments and needs further exploration. We claim that the provision of sophisticated trust evaluation methods in multi-agent environments can overcome the current restraints for a wider acceptance of electronic and mobile commerce applications.

The importance of trust and reputation measurement between autonomous agents or services is widely recognized [12, 13, 15] among researchers. For instance, Sycara [12] and Kollingbaum et al. [13] addressed this problem by introducing a third party which acts as trusted and certificate authority to guarantee the identity of agents and services. Another approach is the rating of a partner after the completion of a transaction following the ebay.com model [15]. This model relies on the availability of a central registry where the global rating for each agent is openly accessible. This simple approach may work well in environments such as online communities or weblogs, but it has several weaknesses when applied to automated electronic and mobile commerce environments. A globally computed trustworthiness value is easily attackable by adding an infinite number of fake ratings to the global value. Furthermore, for the sake of fairness every agent would have to comply to the same regulations on how to rate its peers in a system with a central registry. This approach would not allow personalized and subjective interpretations and calculations of values for trustworthiness and credibility.

We claim that only the individual computation of trustworthiness and credibility values of one agent for another agent can fulfill the demand for security in multi-agent networks. These individual calculations can still use information provided by external sources such as peer nodes or registries, but the evaluation and processing of those information must be based on the individual settings of each agent.

More recent contributions to the evaluation of trust and reputation using Bayesian networks [14] and fuzzy logic concepts [1, 5, 7] provided a starting point to improve the modeling capabilities of social networks. However, these models lack the individual trustworthiness and credibility computation as well as an integrated model which

adjusts and reviews its parameters after a business interaction. They also do not recognize individual security requirements of agent-owners sufficiently.

Bayesian networks, as introduced by [14], provide the required flexibility to evaluate competence and reliability of peer to peer-style interactions between two contracting agents. They demonstrate that the exchange of information about trust and reputation on peer agents increases the performance of the network. Castelfranchi et al. [5] use Fuzzy Cognitive Maps (FCM) [11] to model the relevance of the system inputs before their aggregation. A different approach is the Regret system [7] which integrates fuzzy concepts into the analysis of social networks in electronic marketplaces.

To better incorporate individual perceptions of trust and credibility on both internal and external information sources, we extend existing models which used fuzzy concepts for trust evaluation.

3 Proposed Fuzzy Trust Model

This section introduces a model for the calculation of trust between autonomous agents in a multi-agent environment such as an e-commerce market. The proposed model describes a trust evaluation process implemented by an agent to measure trust in a future negotiation partner before the negotiation process takes place. Furthermore, a model for the review of trustworthiness after the business interaction and a model for the adjustment of the credibility of neighboring agents are introduced in following sections.

The agent to be assessed for trustworthiness will be called *Trusted Agent* and the agent assessing the Trusted Agent's trustworthiness will be called *Trusting Agent*. Peer agents which share information about their past experiences with the Requesting Agent are called *Recommending Agents*. Following [8, 9], we define **Trust** as the belief that the Trusting Agent has in the Trusted Agent's willingness and capability to deliver a quality of service in a given context at a given timeslot. We use the notion trustworthiness, as a measure, to quantify the trust level the Requesting Agent has in the Trusted Agent in a given context at a given time slot.

The Trusting Agent first locates the target service or agent that meets his expectations. It then retrieves information about the Trusted Agent from past interactions from its personal information repository. In addition, the Trusting Agent broadcasts a ReputationRequest to all known agents in his vicinity. Neighboring agents, which have had interaction with the Trusted Agent in the given context, may answer this *ReputationRequest* with a *ReputationResponse* and thus act as Recommending Agents.

The ReputationRequest and the ReputationResponse messages are defined by the following associated ontologies:

ReputationRequest [trusting agent identifier, Trusted Agent identifier, context]

ReputationResponse [recommending agent identifier, Trusted Agent identifier, context, time slot, trustworthiness scale min, trustworthiness scale max, trustworthiness value]

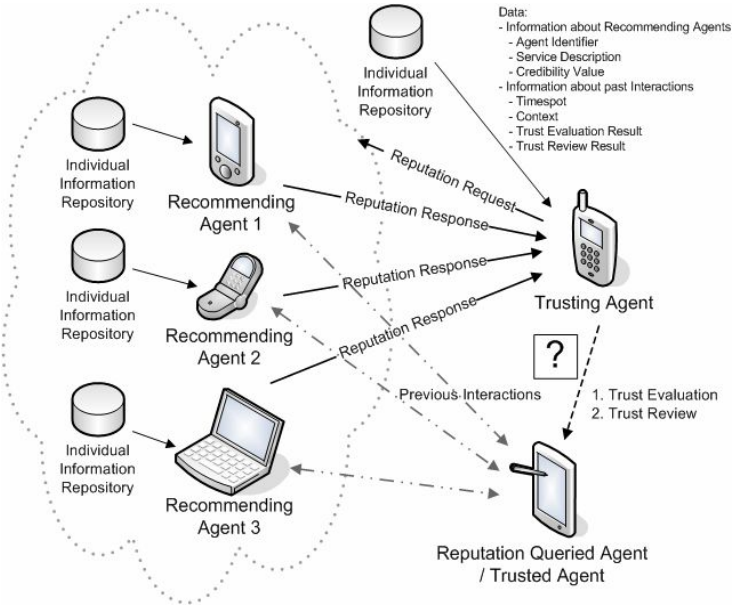


Fig. 1. Social context for trust evaluation model

3.1 Trustworthiness Evaluation

After receiving a message from a number of Recommending Agents, the Trusting Agent has to pre-process the provided information. If a message contains a collection of timeslots and associated trustworthiness values, rather than a single tuple, the data will be aggregated into a single trustworthiness value. This is achieved by sorting the collection (with size *S*) by its timeslots and then calculating an overall *weighted trustworthiness value* (*WTV*) as shown in equation (1). There *n* denotes the current time slot, *m* denotes the timeslot of the following interaction, and *D* denotes a term which characterizes the rate of decay. Furthermore, we scale each trustworthiness value (t_{val}) from the range [trustworthiness scale min (t_{min}), trustworthiness scale max (t_{max})] to our trustworthiness value range of [0,5]. The trustworthiness values are weighted by their timeliness within the time spot collection using an exponential function.

$$\mathrm{WTV} = \frac{\sum_{s=1}^{S}\left[e^{\frac{-(n-m)}{D}} \cdot \left(\frac{t_{val} - t_{\min}}{t_{\max} - t_{\min}} \cdot 5 \right) \right]}{S} \tag{1}$$

Additionally, if the Recommending Agent had multiple interactions with the Trusted Agent and, thus, sends multiple timeslot/trustworthiness values, it provides a more accurate picture in his opinion. This increased accuracy is reflected through the introduction of an opinion weight (*OW*) attribute used in the fuzzy inference engine.

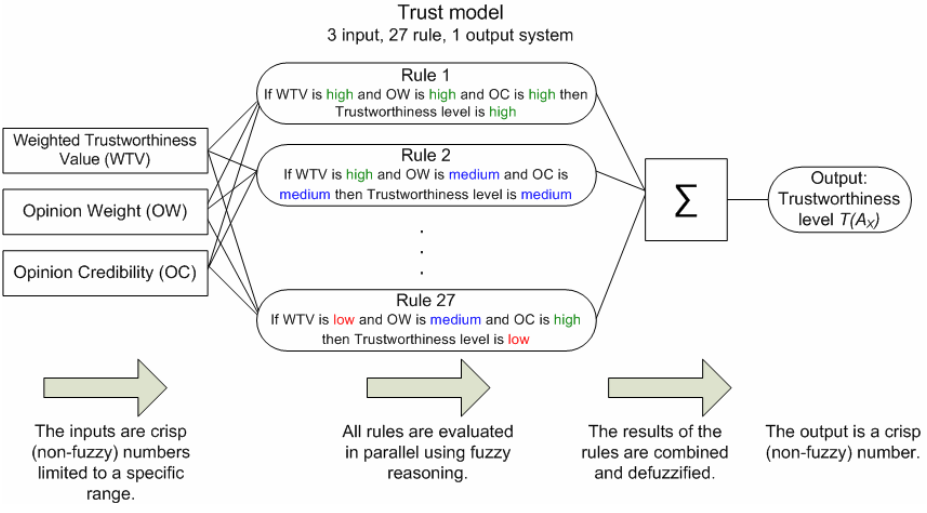


Fig. 2. Conceptual trust model a using fuzzy system [4]

The newly gained information will then be fed into a fuzzy inference engine in order to calculate a trustworthiness value that provides comparable information about the Trusted Agent as reported by the Recommending Agent (see Figure 2).

Prior to the processing of the inputs, it is necessary to create fuzzy membership functions which define the degree of membership of each input parameter in the context of the proposed model. Furthermore, fuzzy rules, based on linguistic variables, which combine the fuzzy sets, are defined in order to characterize the output of the model.

$$OTV\,(A_T) = W_I \cdot T_T + W_E \cdot \frac{\sum_{p=1}^{l} \left(T(A_p)\right)}{l} \tag{2}$$

The linguistic fuzzy rules translate into fuzzy numbers according to the previously created membership functions. These fuzzy numbers then serve as input for the fuzzy expert system. Once these pre-processing steps are accomplished, the process of defuzzification creates the desired crisp overall trustworthiness value for each Recommending Agent ($T(A_P)$). In order to provide additional flexibility, we multiply an internal weight W_I to the trustworthiness query result T_T from the Requesting Agent's internal database. Furthermore, we multiply an external weight W_E to the summarized trustworthiness values of all Recommending Agents to retrieve the overall trustworthiness value ($OTV(A_T)$) for the Trusted Agent where l represents the count of all Recommending Agents who responded to the ReputationRequest.

3.2 Business Interaction Review

To improve his performance and quality of service, the Trusting Agent reviews the business interaction with the Trusted Agent after its completion. There are certain

decisive factors which the Trusting Agent needs to measure to ensure an unbiased assessment of the performance of the Trusted Agent and success of the business relationship. This conclusion not only allows the Trusting Agent to fine-tune its trustworthiness value of the Trusted Agent for future interactions but also allows the adjustment of the credibility values given to each of the Recommending Agents who answered the ReputationRequest previously.

We use the methodology known as CCCI (Correlation, Commitment, Clarity, and Influence) [8, 9] to review the business interaction with the Trusted Agent. The central objective of this methodology is the measurement of the correlation between the service description, both agents agreed to, before their business interaction (expected behavior) and the actually delivered services during the business interaction (actual behavior). The overall correlation measurement is performed through the assessment of three factors which play an important role in the business interaction review process:

1. The commitment to the criterion: The commitment value measures the actual degree of fulfillment of every specified criterion.
2. The clarity of the criterion: The clarity value measures if a service condition was clearly specified, commonly understood, and mutually agreed to. It contains terms and conditions which serve as criteria for the trustworthiness measure.
3. The importance of each criterion: This value measures influence of each criterion as perceived by the Trusting Agent. For example, the agent may find that a complete and timely delivery of the ordered goods or services is more important then the payment method.

After reviewing the commitment, clarity and influence factor for each criterion (c) we combine them with the following expression where N represents the number of all criteria to which the Trusting and the Trusted Agent mutually agreed upon:

$$RC = \frac{\sum_{C=1}^{N} Commit_{criterion(c)} \cdot Clarity_{criterion(c)} \cdot Influence_{criterion(c)}}{5 \cdot \sum_{C=1}^{N} Max_Clarity_{criterion(c)} \cdot Max_Influence_{criterion(c)}} \quad (3)$$

The relative correlation (RC) value is determined by the ratio of the correlation value and the maximum possible correlation value.

3.3 Credibility Value Adjustments

The Recommending Agent has a substantial interest to deliver a truthful opinion to the Trusting Agent because his own credibility and the associated trustworthiness value are at stake. This is because the Trusting Agent will eventually review the given opinion and the actual performance during the business interaction and adjust the credibility value (CV) of the Recommending Agent accordingly.

Therefore to increase the accuracy of the credibility value, we need an adjustment method that allows the reinforcement of the credibility for opinions close to the actual outcome (C_R) of the business interaction. But the method also needs to penalise the credibility value for opinions differing from the actual business outcome C_R. The introduction of a tolerance ε helps to determine whether the credibility value is to be increased for opinions within the pre-defined tolerance or decreased for those outside.

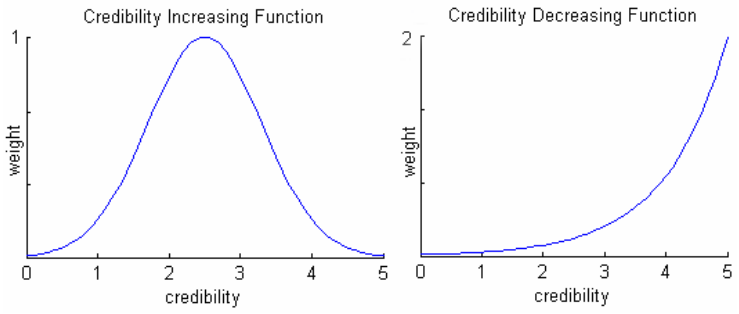


Fig. 3. Functions for Credibility increase, Credibility decrease

If the Trusted Agent behaved inconsistently in the past, the Recommending Agents will deliver very different opinions of the Trusted Agent's performance. Highly differing trustworthiness values delivered from various Recommending Agents can be interpreted in two different ways:

1. Because of the fluctuating performance of the Trusted Agent, most delivered opinions could be truthful.
2. Some of the Recommending Agents may be dishonest in their recommendation to gain an advantage.

Since the Trusting Agent cannot determine the cause for the fluctuations, he reacts in the following way: The more a Trusted Agent's behaviour fluctuates, the lower he adjusts the Recommending Agent's credibility in either direction. The Trusted Agent's fluctuation rate is represented by the *variance* (σ^2) of all opinions (n) delivered by the Recommending Agents. This means the higher the standard deviation the lower our ability of measuring the truthfulness of one opinion.

$$CV_{new} = CV_{old} + \left(1 - \frac{\sigma^2}{CV_{\max}}\right) \cdot (adj) \; ;$$

where:

$$(adj) = \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(CV_{old} - \bar{x})^2}{2\sigma^2}} \right) \quad \text{for } |C_R| \leq \varepsilon$$

$$(adj) = \left[\left(- \frac{e^{CV_{old}}}{e^{CV_{\max}}} \right) \cdot 2 \right] \quad \text{for } |C_R| > \varepsilon \qquad (4)$$

$$\text{and } \sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2 \; ;$$

$$\text{and } \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Also, reflecting the human perception of credibility adjustment, we need to define separate functions for the tasks of credibility reducing and credibility increasing. We choose a bell-shaped function (see Figure 3 (left)) which allows a slow increase of the credibility value if the existing credibility is either relatively low or relatively high and a strong increase of the credibility value if the existing credibility is medium.
For the task of credibility reducing, we choose an exponential function (see Figure 3 (right)) which reduces the credibility slowly if the existing value is already at low and medium levels but decreases the credibility strongly if the existing credibility is close to its maximum. This strategy prevents possible cyclic dishonesty behaviour by Recommending Agents since our exponential function imposes severe punishment in such cases.

4 Evolution of Peer Credibility Records

We examined the development of the credibility values for the Recommending Agents using data from several credibility adjustment cycles recorded during model tests. These cycles result from business interactions within a pre-defined context. Taking into consideration that Recommending Agents follow their own agenda, they might deliberately provide false information. For example, they could deliver correct opinions when the profit or impact of the business interaction is small. A false opinion could be provided if the Recommending Agent was able to gain an advantage.

Our credibility adjustment model reflects this consideration by providing separate functions for increasing and decreasing the credibility values of Recommending Agents. Figure 4 depicts the development of credibility values during our tests. It is clearly visible, that the reduction of credibility values is greater if the trustworthiness review calculations have a negative outcome than vice-versa.

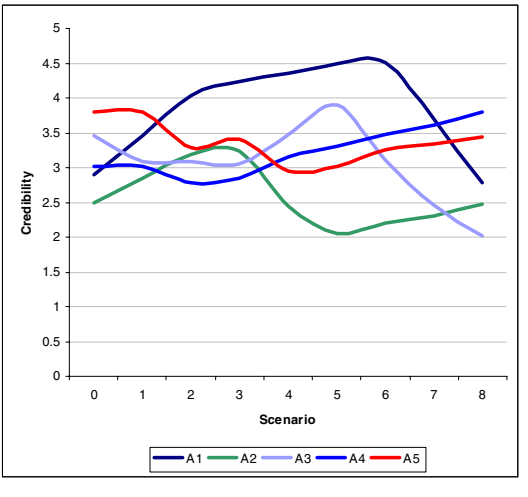


Fig. 4. Development of credibility values

5 Conclusion and Future Work

The ability of an autonomous agent to calculate trustworthiness and credibility values for its peer agents is crucial for the success of automated interactions in multi-agent environments. The characteristics of fuzzy logic, to model uncertainty along with the capability to allow individual perceptions of trust, makes it an excellent methodology to mimic social behaviour in multi-agent environment such as an e-commerce market. This paper has proposed a fuzzy trust model which quantifies and formalises human perception of trust. Furthermore, we introduced a model for the measurement and adjustment of credibility values for peer agents. Our models are highly customisable reflecting the demand for individual setups to meet the security needs of different users.

We have shown how external information, received from peer agents, can be taken into account when evaluating a Trusted Agent. Also, our model provided a mechanism to weight the data set delivered by a Recommending Agent with the individual credibility value for that agent. The Trusting Agent maintains credibility records of the Recommending Agents to prevent possible attacks such as the delivery of false information or cyclic dishonesty. In addition, we accounted for past experiences between the Recommending Agents and the Trusted Agent.

In summary, we examined that during several test-runs our overall model has performed as expected. Both, trustworthiness values and credibility records mature over time, hence increasing the confidence and precision of our model. Future work will validate our model in a large sized e-commerce environment to assess the framework accuracy on a long-term basis. Eventually our model can be incorporated into existing e-commerce markets as a key building block where autonomous agents interact by imitating human social behaviour.

References

1. del Acebo E., de la Rosa, J., L., A Fuzzy System Based Approach to Social Modeling in Multi-Agent Systems, in Proceedings of the first international joint conference on Autonomous agents and multiagent systems, Bologna Italy, 2002.
2. Berthold, M., R.; Hand, D., J., Intelligent Data Analysis, In Chapter 9: Fuzzy Logic p. 321 – 350, Berlin, Springer, 2003
3. Zadeh, L. A., Fuzzy logic, neural networks, and soft computing, in Communications of the ACM archive, Volume 37, Issue 3 Pages: 77 – 84, March 1994
4. The MathWorks, Fuzzy Logic Toolbox for Mathlab Documentation, Available at: http://www.mathworks.com/access/helpdesk/help/toolbox/fuzzy/ (Accessed: 2005, June 16)
5. Castelfranchi, C., Falcone, R., Pezzulo, G., Trust in Information Sources as a Source for Trust: A Fuzzy Approach, in Proceedings of the second international joint conference on Autonomous agents and multiagent systems, Melbourne Australia, 2003.
6. Brenner, W., Zarnekow, R., Wittig, H., Intelligent Software Agents: Foundations and Applications, Springer-Verlag, pp 267-299, 1998
7. Sabater, J., Sierra, C., Reputation and Social Network Analysis in Multi-Agent Systems, in Proceedings of the first international joint conference on Autonomous agents and multiagent systems, Bologna Italy, 2002.

8. Chang, E., Dillon, T., Hussain, F.K., Trust and Reputation for Service-oriented Environments, John Wiley & Sons, to appear Oct. 2005, ISBN: 0-470-01547-0
9. Hussain, F.K., Chang, E., Dillon, T., Trustworthiness and CCCI Metrics for Assigning Trustworthiness in P2P Communication., in Intl. J. Computer Systems Science and Eng. vol. 19, no. 4, pp. 95-112, 2004
10. Schein, A., Popescul, A., Ungar, L., Pennock, D., Methods and metrics for cold-start recommendations, in Proceedings of the 25'th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002), pp. 253-260. Tampere, Finland. August, 2002.
11. Kosko, B. Fuzzy Cognitive Maps, in International Journal Man-Machine Studies, vol. 24, pp.65-75, 1986
12. Sycara, K., Multi-agent Infrastructure, agent discovery, middle agents for Web services and interoperation Multi-agents systems and applications, Springer-Verlag New York, Inc., Pages: 17-49, 2001
13. Kollingbaum, M. J., Norman, T. J., Supervised interaction: creating a web of trust for contracting agents in electronic environments, in Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1, Bologna, Italy, Pages: 272 – 279, 2002
14. Wang, Y., Vassileva, J., Bayesian Network-Based Trust Model, in Proceedings of the IEEE/WIC International Conference on Web Intelligence (WI'0**3**)
15. Resnick, P., Zeckhauser, R., Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System. The Economics of the Internet and E-Commerce. Advances in Applied Microeconomics, 11, 2002.

Building e-Laws Ontology: New Approach

Ahmad Kayed**

Applied Science University, Amman Jordan,
SNC- Monash Melbourne (On-leave), Australia
kayed_a@asu.edu.jo
http://kayed_a.asu.edu.jo

Abstract. Semantic Web provides tools for expressing information in a machine accessible form where agents (human or software) can understand it. Ontology is required to describe the semantics of concepts and properties used in web documents. Ontology is needed to describe products, services, processes and practices in any e-commerce application. Ontology plays an essential role in recognizing the meaning of the information in Web documents. This paper attempts to deploy these concepts in an e-law application. E-laws ontology has been built using existing resources. It has been shown that extracting concepts is less hard than building relationships among them. A new algorithm has been proposed to reduce the number of relationships, so the domain knowledge expert (i.e. lawyer) can refine these relationships.

Keywords: Ontology, e-Laws, Semantic web, E-Commerce, Text-Mining.

1 Introduction

The term "ontology" has its roots in philosophy which has been defined as a particular theory about the nature of being or the kinds of existence [14]. In the computer science community, ontology becomes an important issue. Many research areas study ontology, fields such as Artificial Intelligence (AI), knowledge-based systems, language engineering, multi-database systems, agent-based systems, information systems, etc [7].

Uschold et al. argue that ontology may take a variety of forms, but necessarily it will include a vocabulary of terms, and some specifications of their meaning. This includes definitions and an indication of how concepts are inter-related which collectively impose a structure on the domain and constrain the possible interpretations of terms [21].

Ontology in the AI and database sense is an elaborate conceptual schema of generic domain concepts and relations with constraint or axioms, and with a formal lexicon or concept-dictionary [13]. Guarino [17] defines ontology as an explicit, partial account of a conceptualization. Gruber [6] defines it as an explicit specification of a conceptualization.

** Assistant Prof. in Computer Information System, Applied Science University, Amman-Jordan. On-leave from SNC- Monash Melbourne Australia.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 826–835, 2005.
© Springer-Verlag Berlin Heidelberg 2005

There are many cases were the intersection of the law and cyberspace gets more evident. For example, in the music world copyright laws have long protected owners of intellectual property. The expanding use of the Internet and file-compression technology makes it hard to protect both the band and the owner [15]. In the last decades, using computer assisted legal systems has received more attention with the development of computer technology. There still is a need to solve many problems in the legal domain such as legal modeling, representing rules and cases, interactions, as well as solving semantic issues. Some new systems are trying to solve these issues. For example, the computer assisted legal systems for Bermer[1].

This paper attempts to deploy the ontological concepts in an e-law application. An e-laws ontology has been built using existing resources. It has been shown that extracting concepts is less hard than building relationships among them. A new algorithm is needed to reduce the number of relationships, so the domain knowledge expert can refine them. This paper has been organized as follows: section two gives a background and motivation for building e-laws ontology. Section three explains the ontology life cycle. Section four identifies the main steps in building an e-laws ontology. Section five concludes the paper.

2 Background: Law and Ontology

With the advent of the semantic Web, many e-commerce applications are starting to use ontology in order to solve many semantic problems. It has been argued that beyond software engineering and process engineering, ontological engineering is the third capability needed if successful e-commerce is to be realized.

Laws consist of rules that regulate the conduct of individuals, businesses, and other organizations within society. It is intended to protect persons and their property from unwanted interference from others. The law forbids persons from engaging in certain undesirable activities [3]. Ontology has been used by many researchers in many law-applications. Bruker et al. [2] use it in artificial legal-reasoning. They proposed a number of primitive functions of legal sources and legal knowledge. Boer et al. used ontology for comparing and harmonizing legislation in the European Union [1]. The Information Society of Technology (IST) group used ontology in their e-court system to organize their databases[2]. Mommers uses legal ontology in collaborative workspace applications [16]. Stranieri et al. evaluate legal knowledge based systems using ontology [19]. Visser et al. discussed the usefulness of legal ontologies in the design of knowledge systems. They claim that these ontologies are reusable and these ontologies could enhance the development of legal knowledge systems. They discussed four legal ontologies and investigated how they can be indexed in a library [27] [24]. Zaibert and Smith explore the normatively elements of the ontology of legal and socio-political institutions. They also provided a general ontological framework within

[1] See, for example, the H. Bermer at. www.innoventures.nl.

[2] http://www.cordis.lu/ist/ka1/administrations/home.html

which different institutions of landed property (and of landed non-property) can be contrasted and compared [18]. There are many ontologies in the legal domain, examples of these ontologies are: McCarty's LLD, Stamper's NORMA, Valente's functional ontology, and Van Kralingen and Visser's frame-based ontology [23] [22] [22]. According to Kralingen, ontology has been applied in the representation of several legal domains. Other domains in which the ontology has been tested are penal law, procedural law and civil law in an IT contract and the relevant provisions of the Dutch Civil Code. Most recently, the Imperial College Library Regulations in Germany have been modeled using ontology [25], [26].

The Motivation: Building e-government applications faces the problem of e-commerce laws. This paper builds a standard for e-commerce laws. It has been found that the ontology provides adequate means to represent different legal domains. By using concepts and structures that come natural to lawyers (norms, acts and concepts), models based on the ontology can be easily understood; the demand of clarity is met. The following are benefits from building ontology for e-laws applications:

- Semantic matchmaking.
- Introducing new legislation depending on previous ones.
- Building new intelligent applications using ontology.
- Providing support for e-government applications.
- Building and supporting e-law expert systems.
- Comparing two different legislations.
- Harmonizing two different regulations.
- Building Bi-lingual e-law matchmaking.

The Jordan Case: This paper has been based on building e-government applications in Jordan. The experiences from this case can easily be generalized for any other situation. For the Jordan case, two scenarios have been identified:

Scenario 1: A company plans to invest in Jordan. It sends its agents (soft or human) to check the Jordan laws. They check the laws of: taxes, investments, copyrights, labors, contracts, etc. A matchmaking is needed for these agents. This matchmaking should include semantic, relational, bi-lingual, and keyword matches.

Scenario 2 : An online court has been held. The lawyers are looking for certain laws and law-cases that can help them. They usually use keyword matching to find these law cases. Jordan, as many countries do, uses their own concepts to define these law cases. There is a need for building a semantic and bi-lingual matching. Through a law ontology these problems can be solved.

3 The Ontology Life Cycle

Building systems using a predefined standard is less complicated than building systems using ontology. The former is simple but not applicable an in open

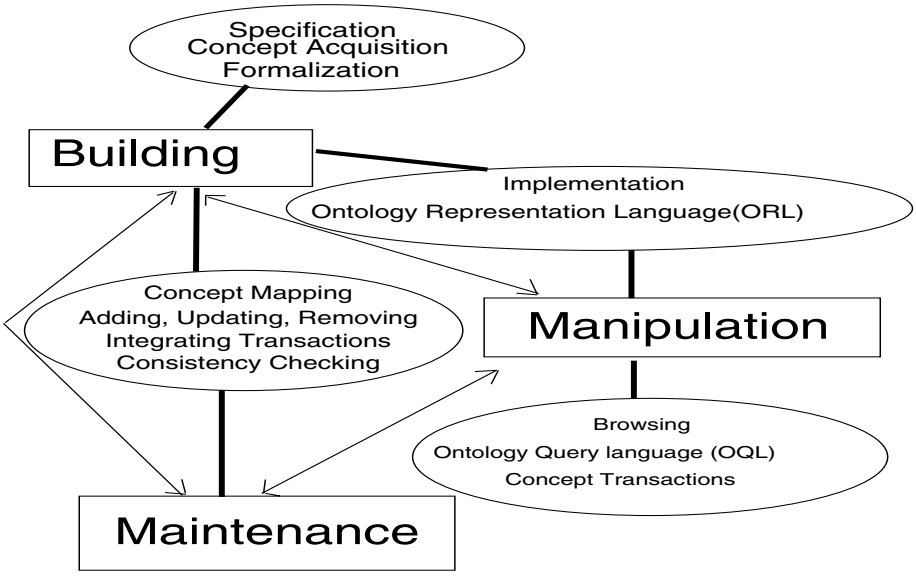


Fig. 1. Ontology Life Cycle

environment. The latter is complex but will move the complexity from the user side to the application building side. This move is needed since we build an application only once but use it many times.

Kayed et al. [10] summarize the methodologies for building ontologies around three major stages of the ontology life cycle i.e. Building-Manipulating-Maintaining (see figure 1).

Building Stage: There are many attempts to define a methodology for ontology construction. Examples are [5] [4] [20]. In the building stage, four steps are needed: specification, conceptualization, formalization, and implementation.

Manipulation Stage: In the manipulation stage, an ontology query language should be provided for browsing and searching; efficient lattice operation; and domain specific operations.

Maintenance Stage: In the maintenance part, developers should be able to syntactically and lexically analyze the ontology, adding, removing, modifying definitions, and also translate from one language to another. Ontologies should be built with different levels of generality. This means that different types of updating mechanisms should be provided. For more detail see [10].

4 Building e-Laws Ontology

Kayed et al. introduced the concept of layered ontologies [11]. Defining levels of abstractions facilitates the process of transforming existing resources to on-

tology. Ontology serves as an abstract data type for concepts in domain. Building a new ontology from scratch is not a simple task. In the concept acquisition process, we need to re-use existing resources to build our ontology. The aim of the concept acquisition process is to facilitate the acquisition, classification, and representation of concepts. Activity tasks include: classification criteria, question scheme, optimization of questions, rules for classifying relations and concept representation [12]. If we agree that ontology is an explicit specification of conceptualization [6], we also need to agree that this knowledge is implicit in many applications. This knowledge may be abstracted from existing resources.

In this paper, an experiment has been conducted to extract concepts for e-law ontology. Text-mining tools have been used to extract concepts in the domain of e-commerce laws. The following summarizes the steps to build this ontology:

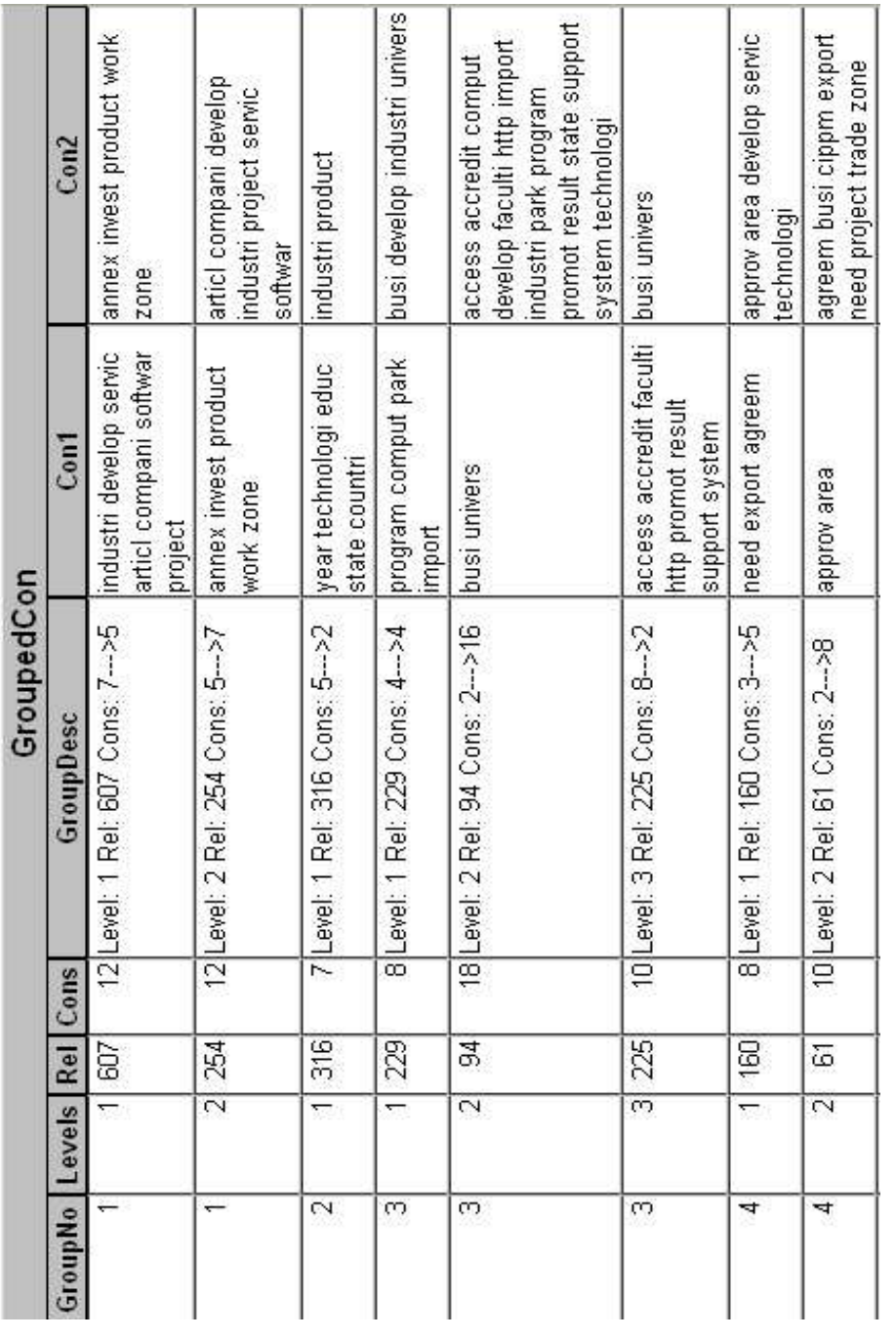
- Collect many law cases for e-commerce.
- Extract top concepts.
- Refine the results.
- Categorize the concepts.
- Define the relationships among concepts.
- Build the ontological hierarchy.
- Formalize the concepts.

Collect Law Cases Form e-Commerce Law Resources: REACH is the main source for our e-law ontology. REACH is a core group of members of the IT industry, supported by the AMIR Program (Access to Microfinance & Improved Implementation of Policy Reform - United States Agency for International Development). REACH devised a strategy and action plan initiative for e-commerce. This initiative was conducted through an intensive consultation and research process with Jordanian IT industry leaders , in addition to international and domestic consultants[3]. The REACH team has developed a framework for e-commerce laws. This paper uses their results and some law cases related to E-commerce. Four cases have been selected. All these resources have been converted to text format. The original size of these documents was around 1 MB. The converted files have been reduced into 367KB.

Extract Top Concepts: KAONs texttoont program has been used[4] to extract the terms and their relationships. Around 273 concepts have been extracted in this stage. The following is just a sample(for detail see [8]):

industri, develop, service, articl, compani, softwar, project, year, technologi, work, educ, state, countri, program, product, comput, import, park, need, export, invest, busi, agreem, tax, inform, law, sector, system, base, govern, support, qual, certif, annex, applic, firm, manag, reach, univers, custom, purpose, skill, local,

[3] http://www.intaj.net/resource.cfm#top
[4] http://kaon.semanticweb.org



GroupedCon

GroupNo	Levels	Rel	Cons	GroupDesc	Con1	Con2
1	1	607	12	Level: 1 Rel: 607 Cons: 7--->5	industri develop servic articl compani softwar project	annex invest product work zone
1	2	254	12	Level: 2 Rel: 254 Cons: 5--->7	annex invest product work zone	articl compani develop industri project servic softwar
2	1	316	7	Level: 1 Rel: 316 Cons: 5--->2	year technologi educ state countri	industri product
3	1	229	8	Level: 1 Rel: 229 Cons: 4--->4	program comput park import	busi develop industri univers
3	2	94	18	Level: 2 Rel: 94 Cons: 2--->16	busi univers	access accredit comput develop faculti http import industri park program promot result state support system technologi
3	3	225	10	Level: 3 Rel: 225 Cons: 8--->2	access accredit faculti http promot result support system	busi univers
4	1	160	8	Level: 1 Rel: 160 Cons: 3--->5	need export agreem	approw area develop servic technologi
4	2	61	10	Level: 2 Rel: 61 Cons: 2--->8	approw area	agreem busi cippm export need project trade zone

Fig. 2. Table 1

corpor, implement, regul, establish, total, time, build, duti, avail, cost, process, employ, organ, train, number, market, activ, zone, case, increase, etc.[5].

Refining Terms and Re-defining Relationships: KAON has the ability to build the relationships for the extracted concepts. KAON algorithm extracts around 32,000 relationships for the 273 concepts. The size of the file in text format goes beyond 10 MB. It was very hard to discuss these relationships with any domain expert (lawyers). A new algorithm has been developed to reduce these relationships to an acceptable number[6].

The main objective of the algorithm is to group the concepts in a way that the domain expert can look at them. The algorithm will find the group of concepts that have at least one relation with another group. All concepts in the first group must have a relationship with the second one. The algorithm converges since the increasing of elements in the first group will decrease the number of elements in the second one. The elements in the first group will be chosen by ordering the concept according to the number of relationships that they have. The same steps will be repeated for a second level. In the second level, we look at the second group and group all the concepts that have a relation with all concepts in the second group. The complexity for the algorithm is liner.

This way works well with our e-law ontology domain. The number of relationships has been reduced from 6000 to 39 groups of relationships with two or three levels (see table 1). This enables the domain concepts (lawyers) to define the types of relationships that relate each group. After that, they can define the relationships that relate each element with another element. The number of elements in each group is critical. Increasing the number of elements in the first group will reduce the number of elements in the second group. However, decreasing the first group will increase the second group. The algorithm has been enhanced by running another algorithm that finds the best number of elements in each group. It has been found that the number of elements in each group should be close to each other. It has been shown experimentally, that best number of elements in each group should be from three to five elements. For more details see [8].

The initial concepts, relationships, and algorithms are available via [8]. KAON has been used to extract the concepts from text format files. MS Access and MS Visual Basic have been used to implement the algorithms. The following are the main steps in the algorithm. The full details of the algorithm will be published in another forum.

Start:

1-Sort the concepts according to the concept with higher number of relationships.

[5] KAON extracts the common terms not the English terms. For example: the concept (develop) is part of developing, development, developed, etc.

[6] The full details of the algorithm will be published in another forum. Here, only the main steps are listed.

2-Group the top concepts. The best top-concept group is defined by another algorithm. The group size varies from three to five elements. This group is named the top group.

3-Find all concepts that has a relationship(s) with all concepts in the top group. Call this the second group.

4-For another level, find all concepts that has a relationship(s) with all concepts in the second group.

5-Remove all concepts from the top group.

6-Repeat step 2-5 until you get an empty set.

5 Conclusions

Building ontology is moving from being craft to being science. The paper, used existing resources(data and tools) to build ontology for e-commerce applications in the domain of e-laws. E-commerce laws have been collected, existing text-mining techniques have been used, basic concepts and relationships have been identified. The number of concepts and the relationships were very huge. A new algorithm has been proposed to reduce the number of relationships and group them with different levels. Concepts have been grouped according to the number of relationships. Each group is linked with other group if and only if each element in the first group has at least one relationship with the second group.

Extracting concepts is not a hard task. Defining relationships for an ontology still depends on the domain expert and needs much more effort. This new algorithm will help the domain expert in defining and refining these relationships.

We are now looking to implement this ontology using Ontology Web Language(OWL), and to build relational matching which we defined in [9]. This will enable and support the reasoning process in many law expert systems.

Acknowledgment

This work has been supported by the school of CIS at the Applied Science University. The author likes to thank many anonymous people for their efforts to improve the readability of this paper. I would also like to thank Hanin for her assistance in proofreading the paper.

References

1. Alexander Boer, Tom M. van Engers, and Radboud Winkels. Using ontologies for comparing and harmonizing legislation. In *ICAIL*, pages 60–69, 2003.
2. Joost Breuker, Andre Valente, and Radboud Winkels. Legal ontologies: A functional view.
3. Henry R. Cheeseman. *Business Law.* Prentice Hall; 5 edition, 2003.
4. M. Fernandez, A. Gomez-Perez, and J. Sierra. Building a chemical ontology using methodology and the ontology design environment. *IEEE Intelligent Systems*, 14,1:37–46, 1999.

5. A. Gomez-Perez and D. Rojas-Amaya. Ontological reengineering for reuse. *Lecture Notes in Computer Science*, 1621:139–149, 1999.
6. Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43(5,6):907–928, 1995.
7. N. Guarino. Formal ontology and information systems. In *Proc. of the 1st International Conference on FOIS, Trento, Italy*, 6-8 June 1998.
8. Ahmad Kayed. *Home Page. http* : *//Kayed_a.asu.edu.jo*, 2005.
9. Ahmad Kayed and Robert Colomb. Re-engineering approach to build domain ontologies. *Web Intelligence: Research and Development: Lecture Notes in Artificial Intelligence(LNAI)*, 2198:464–472, 2001.
10. Ahmad Kayed and Robert Colomb. Extracting ontological concepts for tendering conceptual structures. *Data and Knowledge Engineering*, 40(1):71–89, 2002.
11. Ahmad Kayed and Robert M. Colomb. Ontological and conceptual structures for tendering automation. In *the Eleventh Australasian Conference on Information Systems (ACIS2000), BRISBANE, AUSTRALIA, ISBN 1 86435 512 3*, 6-8 December 2000.
12. Maria Lee, Kwang Mong Sim, Paul Kwok, and Gabriel Chau. An ontology-based approach for e-commerce concept acquisition. In *The Australian Workshop on AI in Electronic Commerce, conjunction with the Australian Joint Conference on Artificial Intelligence (AI'99) Sydney, Australia, ISBN 0643065520*, pages 103–107, Dec. 1999.
13. Fritz Lehmann. *Machine-Negotiated, Ontology-Based EDI In Lecture Notes in Computer Science: Electronic Commerce, Current Research Issues and Application by Nabil R. Adam and Yelena Yesha*. Springer 1028, 1995.
14. Incorporated Merriam-Webster. *WWWebster Dictionary*. http://www.m-w.com/dictionary, 1999.
15. Roger LeRoy Miller and Gaylord Jentz. *Fundamentals of Business Law*. West Legal Studies in Business, 5th Edition, 2005.
16. Laurens Mommers. Applications of a knowledge-based ontology of the legal domain in collaborative workspaces. In *ICAIL*, pages 70–108, 2003.
17. Guarino N. Semantic matching: Formal ontological distinctions for information organization, extraction, and integration. *In M. T. Pazienza (ed.) Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology, Springer Verlag*, 1299:139–170, 1997.
18. B. Smith and L. Zaibert. The metaphysics of real estate, 1997.
19. Andrew Stranieri and John Zeleznikow. The evaluation of legal knowledge based systems. In *International Conference on Artificial Intelligence and Law*, pages 18–24, 1999.
20. Mike Uschold. Building ontologies: Towards a unified methodology. In *16th Annual Conf. of the British Computer Society Specialist Group on Expert Systems*, Cambridge, UK, 1996.
21. Mike Uschold and Robert Jasper. A framework for understanding and classifying ontology applications. In V. Benjamins, B. Chasdrasekaran anf A. Gomez-Perez, N. Guarino, and M. Uschold, editors, *Proceedings of the IJCAI-99 workshop on ontologies and Problem-Solving Methods (KRR5)*, volume 18, pages 1–11, Stockholm, Sweden, Aug. 2 1999. CEUR-WS.
22. Robert W. van Kralingen, Pepijn R. S. Visser, Trevor J. M. Bench-Capon, and H. Jaap van den Herik. A principled approach to developing legal knowledge systems, 1999.

23. P. Visser. The formal specification of a legal ontology, 1996.
24. P. Visser. A comparison of two legal ontologies, 1997.
25. Pepijn R. S. Visser and Trevor J. M. Bench-Capon. On the reusability of ontologies in knowledge system design. In *DEXA Workshop*, pages 256–261, 1996.
26. Pepijn R. S. Visser, Robert W. van Kralingen, and Trevor J. M. Bench-Capon. A method for the development of legal knowledge systems. In *International Conference on Artificial Intelligence and Law*, pages 151–160, 1997.
27. Pepijn R.S. Visser and Trevor J.M. Bench-Capon. Ontologies in the design of legal knowledge systems ; towards a library of legal domain ontologies.

Generation of Standardised Rights Expressions from Contracts: An Ontology Approach?

Silvia Llorente[1], Jaime Delgado[1], Eva Rodríguez[1], Rubén Barrio[1], Isabella Longo[2], and Franco Bixio[2]

[1] Universitat Pompeu Fabra, Passeig de Circumval·lació, 8 - 08003 Barcelona, Spain
[2] Associazione dei Fonografici Italiani, Via Vittor Pisani 10 - 20124 Milano, Italy
{silvia.llorente, jaime.delgado, eva.rodriguez, ruben.barrio}@upf.edu,
{europeanbranch, francobixio}@afi.mi.it
http://dmag.upf.edu, http://www.afi.mi.it

Abstract. Distribution of multimedia copyrighted material is a hot topic for the music and entertainment industry. Piracy, peer to peer networks and portable devices make multimedia content easily transferable without respecting the associated rights and protection mechanisms. Standard and industry-led initiatives try to prevent this unauthorised usage by means of electronic protection and governance mechanisms. On the other hand, different organisations have been handling related legal issues by means of paper contracts. Now, the question is: How can we relate electronic protection measures with the paper contracts behind them?

This paper presents an analysis of current contract clauses and an approach to generate, from them, standardised rights expressions, or licenses, in an as automatic as possible way. A mapping between those contract clauses and MPEG-21 REL (Rights Expression Language), the most promising rights expressions standard, is also proposed, including an initial relational model database structure. An ontology-based approach for the problem is also pointed out. If contract clauses could be expressed as part of an ontology, this would facilitate the automatic licenses generation. Generic contracts ontologies and specific intellectual property rights ontologies would be the starting point, together with the presented analysis.

1 Introduction

Copyright information for the distribution of multimedia material is, in practice, usually expressed in paper contracts, which define the framework of usage of this material between the parties involved in the contract. This is specially important in the music industry, where many parties can be involved and a piece of music can have many different usages: addition to a film or an advertisement, public performance, distribution in the digital market, etc.

Control of copyrighted multimedia material usage is becoming more and more important because of piracy and the easy distribution of this material by peer-to-peer networks and other electronic means. In this environment, using paper contracts for implementing governance of multimedia content is not feasible and electronic means

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 836 – 845, 2005.
© Springer-Verlag Berlin Heidelberg 2005

for providing protection and governance mechanisms are needed. These mechanisms are being currently defined by standardisation and industry initiatives, like MPEG-21 [1], OMA [2], DMP [3] or Microsoft [4]. They propose mechanisms for protecting and governing multimedia content, allowing automatic control of content usage. It is also worth mentioning the work being done in the Creative Commons project (http://creativecommons.org/), that has created a number of licenses and semantics for describing usage rights over content.

Nevertheless, we still have the problem of how we can establish the connection from what is described in the paper contract and the electronic protection and governance means in a comprehensive and feasible way.

In this paper we propose an initial approach to this problem, based on the study of the clauses present in current music industry contracts and the relationship of these clauses with a standardised rights expression language defined by MPEG-21: MPEG-21 REL [5]. Other parts of this standard, IPMP [6] and RDD [7], are also involved in the protection and governance of multimedia content as explained throughout the following sections.

Based on the work we have done in the area of intellectual property rights (IPR) ontologies [8], and the work done elsewhere on contract ontologies [9], we could identify an ontology-based approach for better achieving our objectives.

2 Extraction of Clauses from Current Contracts

Nowadays, there are many kinds of contracts in the music and entertainment industry. As we have already mentioned, these contracts are mainly in paper and usually define a framework of how two companies want to work together.

Our first aim is to study the clauses existing in current contracts in order to define its structure using a relational database. This will facilitate the semi-automated extraction of clauses from current contracts and also the creation of new contracts and licenses, as we will explain later.

The first extractions should be done by a person, before any automation. Then, by implementing tools using pattern matching techniques, other contracts could be analysed and the clauses already stored in the system could be recognised. A human operator should check that the found clauses are correct, in order to add them to the contracts database. Missing clauses should be added by the human operator, in order that they could be found in further uses of the tool.

Another possible approach for the contract description could be the use of ontologies. In this way, we could define contract structure, including clauses, participants and other elements and their relationships, being able to use other related ontologies defining contractual terms and conditions applicable to the corresponding sector.

In our research group, DMAG [10], we have already worked with ontologies for describing IPR and rights associated to multimedia content. We developed IPROnto [8], an ontology for IPR [11]. We have continued this work, by defining RDDOnto [12], that is an ontology that translates the MPEG-21 RDD (Rights Data Dictionary) specification into a hierarchical set of definitions with semantic content included. RDDOnto translates the RDD specification into a machine-readable semantic engine that enables automatic handling of rights expressions.

Other initiatives in the definition of ontologies for contract representation exist. For instance, [9] mentions the concept of upper layer for contract ontologies. This upper layer could be used as a basis to describe any kind of contract, and, specifically, those related to entertainment and music industry, that are the ones of our particular interest.

3 Relationship Among Contracts, Rights Expressions and Clauses

We want to make several uses of the clauses extracted from contracts. The first one is to extract clauses and generate relationships among them with other contracts, as explained in section 2. The second one is to generate new contracts and rights expressions from these clauses.

The objective of generating new contracts and rights expressions is to help users in the creation of new contracts that are formed by the clauses found. The more clauses recognised, the more kinds of contracts that can be created. This will be very helpful for the process of creating contracts and rights expressions derived from them, that is usually very complex for those that are not experts in legal issues.

Nevertheless, contracts and rights expressions creation is not the only use we can make of the clauses extraction functionality. The ability to create standardised rights expressions from contracts gives us the possibility of interchanging contract information in a standardised way, allows verification of contracts by using automatic and standardised mechanisms and may improve the understanding of legal clauses.

4 Description of a Relational Model of Standardised Rights Expressions

Standardisation of rights expressions, specially those that describe usage and distribution of multimedia content, is currently an important issue that involves both industry and standards bodies.

There are mainly two initiatives for the description of rights expression languages: MPEG-21 REL and ODRL [13]. Both languages follow the same axiomatic principles, although they slightly differ in the functionality they provide. This similarity could permit working with both of them without major inconveniences. We are currently working with both languages, implementing tools and performing research about their interoperability [15], [16]. However, we have first considered MPEG-21 REL to define a relational model for representing rights expressions.

This section briefly describes MPEG-21 and the relational model we have defined for storing licenses expressed in MPEG-21 REL.

4.1 MPEG-21

The MPEG-21 standard defines different mechanisms and elements needed to support multimedia information delivery and management, and the relationships and operations supported by them. In the different parts of the standard, these elements are elaborated by defining the syntax and semantics of their characteristics.

In the MPEG-21 context, the information is structured in Digital Items, which are the fundamental unit of distribution and transaction. A Digital Item [14] is constituted by the digital content, plus related metadata, such as information related to the protection tools (IPMP, Part 4) [6] or rights expressions (REL, Part 5) [5].

The protection and governance of digital content are specified in IPMP Components, REL and RDD MPEG-21 parts. IPMP Components provides mechanisms to protect digital items and to associate rights expressions to the target of their governance, while MPEG-21 REL specifies the syntax and semantics of the language for issuing rights for users to act on digital items. Finally, MPEG-21 RDD (Part 6) [7] comprises a set of terms to support the MPEG-21 REL.

As we have already mentioned, MPEG-21 REL specifies the syntax and semantics of a Rights Expression Language (REL). RELs have been proposed to express rights and conditions of use of digital content and can be used for example to describe an agreement between a content provider and a distributor, or between a distributor and an end user. Moreover, RELs can be used to express the copyright associated to a given digital content by specifying under which conditions the user is allowed to exercise a right.

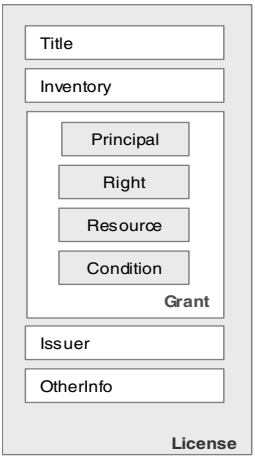


Fig. 1. REL License

MPEG-21 REL defines the License, a container of grants that are formed by a principal that has the permission to exercise a right against a resource under some conditions that must be previously fulfilled. Its structure is shown in Figure 1.

Inside a REL license, the most important element is the Grant. A Grant is an XML structure that is formed by four elements:

- Principal represents the unique identification of an entity involved in the granting or exercising of Rights.
- Right specifies an action or activity that a Principal may perform on, or using, some associated Resource.

- Resource represents the object against which the Principal of a Grant has the Right to perform.
- Condition represents grammatical terms, conditions and obligations that a Principal must satisfy before it may take advantage of an authorisation conveyed to it in a Grant.

A Grant expresses that some Principal may exercise some Right against some Resource, subject, possibly, to some Condition.

MPEG-21 REL makes use of the Rights Data Dictionary, part 6 of the MPEG-21 standard, that comprises a set of clear, consistent, structured, integrated and uniquely identified terms. The structure of the RDD is designed to provide a set of well-defined terms for use in rights expressions.

4.2 Relational Model for Licenses

Figure 2 shows the entity relationship model we have defined for storing MPEG-21 REL XML-based licenses into a relational database. Some of the fields, specially those expressing conditions, are XML structures that have to be parsed after getting them from the database.

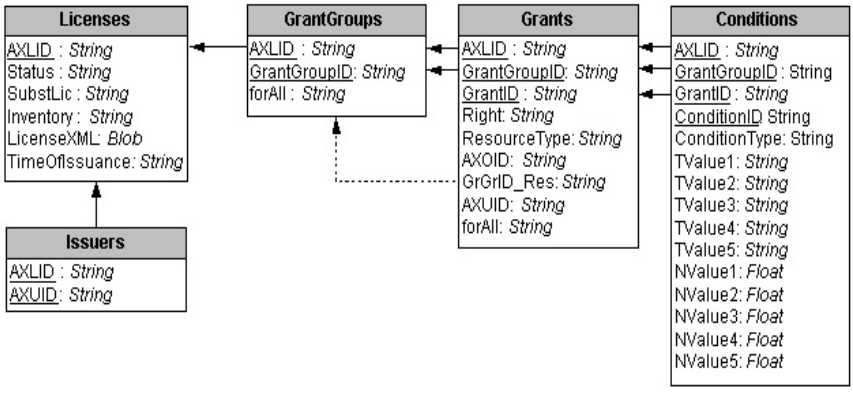


Fig. 2. Entity relationship model for MPEG-21 REL licenses

The main aim of using a relational database for modelling licenses is to improve license searches in order to authorise operations over content to users, applying the authorisation model described in MPEG-21 REL standard.

The description of a common relational model that may be followed by different rights expression languages also facilitates the translation of rights expressions between the languages selected [15], [16], authorisation [17] using licenses described in different languages or even those generated from paper contracts.

5 License and Contracts Analysis

This section presents an analysis of current contracts and licenses, describing their structure and their main elements. Then, licensing models for the music sector will be outlined and compared with rights expressions described in MPEG-21 REL.

5.1 Elements in Licenses and Contracts

The assignment of copyright is essentially a transfer of rights, that is, a sale of rights. In this way, a rights holder should be aware that assigning rights usually implies that ownership (and possibly control) is transferred. In order to allow usage of rights but maintaining the ownership and control, licensing of rights is used.

A license agreement may have the following elements:

- Parties describe the persons or companies involved in the contract.
- Recitals describe the purpose of the contract, but they are not really part of it.
- Definitions element is very important from a legal point of view as precise wording is required. Thus, where concepts are complex or it might take some time to explain it in a short phrase, a word is chosen as shorthand to signify them. Any change in the meaning of a definition can have a significant impact through the whole contract.
- Agreement's purpose is the main part of the contract and summarises what is being provided for the price. Anything which is not included on the agreement should be separately negotiated for an extra fee.
- Rights granted under the license.
- Usage restrictions define what cannot be done with the licensed materials.
- Territory describes where the material will be exploited. This allows the licenser to license or retain rights over certain territories.
- Term defines duration of the contract.
- Exclusivity – non-exclusivity, defines if it is possible that several parties exercise the same rights or not within the same territories.
- Delivery and access to the licensed materials has to be described in the contract in order to avoid later dispute. It includes the approximate date of delivery, the format, the media, etc.
- License fee depends on the circumstances and it could be a royalty, a fee, a combination of both or other possibilities. In any case, the choice of the fee (amount and condition) depends on the sort of deal.
- Licensee's undertakings and obligations. When included, they compromise licensee or his users will not infringe copyright or proprietary rights of the licensed materials. Moreover, it also compromises the licensee and his users to use licensed materials according to the license.
- Warranties and indemnities. A warranty is a statement or representation that certain facts are true. For instance, a warranty may include that the goods and/or services will perform as promised in the agreement. An indemnity is one party's agreement to insure or compensate the other party against losses and expenses resulting from failures in performance under the contract.
- Force majeure is a condition beyond the control of the parties such as war, strikes, destruction of network facilities, etc. not foreseen by the parties and which prevented performance under the contract.
- Assignment and subcontracting elements gives the licensee the right to assign rights under the contract to third parties. As assignment cannot be easily done in most jurisdictions, many licenses clearly indicate that it cannot be performed without the prior written consent of the other party.

- Jurisdiction defines the applicable state law of the contract.
- Signature of the corresponding parties.

5.2 Licenses in the Music Sector

In the previous section we have described the main elements of a contract or license. As all these elements may not be present in a specific scenario, we describe here the ones that are present in music sector. Nevertheless, inside this sector, licenses can be different depending on the licensor. Although we have analysed more cases, we just present the case where the licensor is a music publisher.

Table 1 describes values for the main fields for this case. The rights that are granted to the licensee depend on the purpose of the contract. Also other conditions are described, like exclusivity, term and territory.

Table 1. Scheme of contracts elements when the licensor is a music publisher

Licensee	Purpose	Granted Rights	Exclusivity	Term	Territory
Distributor	Sell the lyrics / sheets	Print, publish, sell	Yes	3 years	One
Sub-publisher	Foreign Country market exploitation	Print, publish, sell; all rights under copyrights	Yes	3 years	One / More
Aggregator	Digital market distributor	Sell, copy, sublicense, distribute	No	5/7 years	World
CD Rom / DVD	Use of text of sheet / lyric	Manufacture, sell, distribute	No	3/5 years	One
On-line retailer	Sell	Sell, copy, sublicense, distribute, transmit	No	1/3 years	World
Film producer	Syncro	Syncro with film and related uses	No	Perpetuity	World
Advertising	Syncro	Syncro connect only with the spot	No	1 year	One
Multimedia CD Rom	Syncro	Syncro connect with other media	No	Perpetuity	One
Merchandising	Sell	Use lyric / sheet music for manufacturing, advertise, distribution, sell	Yes	1 / more years	One / world
Publisher	Use of text of lyric / sheets in a book	Print lyric / sheet in a book or magazine, sell, distribution	No	Perpetuity	One / world

Table 2. Equivalences between contracts and MPEG-21 REL licenses

Contracts		MPEG-21 REL Licenses
Contracting parties		Principals, Issuers
Media in which to use the licensed work		ExerciseMechanism condition
The degree of exclusivity involved in the license or other grant of rights		Could be expressed in the otherInfo element of the license, if this element is only informative
The term or duration of the license or other grant		Validity Intervals
The territory for which the rights are granted		Territory
The specific rights granted		Rights in the RDD
Any reserved rights		If rights not granted, then reserved (no need to specify them in the license)
The remuneration method	Flat fees	Fee flat
	Advances (payments which apply against royalties)	Fees prepay
	Royalties	Fees
	Option payments	FeePerUse, feeFlat, feePerInterval, feeMetered, feePerUsePrepay. All with bank account, payment service, …
The credit or billing to the author or owner of the underlying work		ExerciseMechanism condition that forces to send an Event Report to the owner of the underlying work that includes the billing
Intellectual property rights enforcement		REL/RDD & IPMP
Exclusive Rights	Reproduction rights	Render, play, …
	Distribution rights	Issue, obtain
	Adaptation rights	Adapt, modify, enhance, reduce, …
	Public Performance rights	Express, perform, …
	Public Display rights	Play, …
	Public Communication rights	Say, express, …
Private copying		Right: Copy (adapt without constraints) Conditions: ExerciseLimit: 1 Destination: one of the user devices
Public Domain		ExerciseMechanism condition
Exclusivity		Requires the definition of a new condition
WCT and WPPT key provisions (rights)	Right of communication to the public	Perform
	Reproduction right	Render, play
	Right of distribution	Issue

5.2 Relationship Between Music Contracts and MPEG-21 REL Licenses

At first sight, the common point between music contracts and MPEG-21 REL licenses are the rights and the term and territory conditions. It is also possible to describe a fee condition on an MPEG-21 REL license, as several of them are considered.

Taking these commonalities into account, a reasonable approach should be the creation of license templates with the usual conditions for them. In the case of conditions like exclusivity of rights transfer, that are not currently considered in the MPEG-21 REL standard, it should be needed to extend the current XML Schema to include this contract clause.

Table 2 summarises the equivalences between the elements required in contracts and the elements currently defined in MPEG-21 REL. Other parts of MPEG-21 involved in the protection and governance of content, like IPMP and RDD are also indicated in the table.

6 Conclusions

Copyrighted multimedia material is licensed between parties by means of contracts. These contracts, mainly in paper format, are expressed by means of clauses. Trying to control what is being expressed in these contracts in the digital world is a very complex issue to solve. For this reason, we propose the extraction of clauses present in contracts and its description using standardised rights expressions, like the ones defined by MPEG-21 REL.

In previous work in our research group DMAG [10], we have implemented several tools [18] that deal with MPEG-21 REL, including creation and validation of licenses and authorisation of multimedia content usage based on the authorisation model described in this standard. This is why we believe that connecting a standard rights expression language with current contracts clauses will facilitate the way contracts in the music, and by extension the multimedia sector, will be created, validated and transformed into machine-readable formats, more suitable for interchange, translation and automation of control processes.

Finally, we have also been working in the description of ontologies for expressing MPEG-21 parts 5 (REL) and 6 (RDD) [12], [19]. This work, together with the contract analysis and clause extraction, could give as a result the creation of a licenses ontology based on contract clauses, permitting the use of other existing ontologies, thus widening its scope.

Acknowledgements. This work has been partly supported by the Spanish administration (AgentWeb project, TIC 2002-01336) and is being developed within VISNET (IST-2003-506946, http://www.visnet-noe.org), a European Network of Excellence and AXMEDIS, a European Integrated Project, both funded under the European Commission IST FP6 program.

References

1. MPEG 21, http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm
2. Open Mobile Alliance (OMA), http://www.openmobilealliance.org

3. Digital Media Project (DMP), http://www.dmpf.org/
4. Microsoft DRM, http://www.microsoft.com/windows/windowsmedia/drm/default.aspx
5. ISO/IEC, ISO/IEC IS 21000-5 - Rights Expression Language
6. ISO/IEC, ISO/IEC CD 21000-4 - Intellectual Property Management and Protection
7. ISO/IEC, ISO/IEC IS 21000-6 - Rights Data Dictionary
8. IPROnto, http://dmag.upf.edu/ontologies/ipronto
9. Kabilian, V., Johannesson, P., Rugaimukamu, D.M.: Business contract obligation monitoring through use of multi tier contract ontology. In: WORM CoRe 2003, LNCS 2889.
10. Distributed Multimedia Applications Group (DMAG), http://dmag.upf.edu
11. Delgado, J., Gallego, I., Llorente, S., García, R., Regulatory Ontologies: An Intellectual Property Rights approach. In: WORM CoRe 2003, LNCS 2889.
12. Delgado, J., Gallego, I., Garcia, R.: Use of Semantic Tools for a Digital Rights Dictionary. In: EC-Web 2004, LNCS 3182.
13. Open Digital Rights Language (ODRL), http://odrl.net
14. ISO/IEC, ISO/IEC 2nd Edition FCD 21000-2 – Digital Item Declaration
15. Prados, J., Rodríguez, E., Delgado, J.: Interoperability between different Rights Expression Languages and Protection Mechanisms. In: AXMEDIS 2005. IEEE CS Press.
16. Prados, J., Rodríguez, E., Delgado, J.: A new Approach for Interoperability between ODRL and MPEG-21 REL. In: Second International ODRL Workshop (2005).
17. Rodríguez, E., Llorente, S., Delgado, J.: Use of rights expression languages for protecting multimedia information. In: WEDELMUSIC 2004 Proceedings, IEEE CS Press.
18. DMAG MPEG 21 Tools, http://dmag.upf.edu/DMAGMPEG21Tools
19. Gil, R., Garcia, R., Delgado, J.: An interoperable framework for IPR using web ontologies. In: LOAIT Workshop 2005, http://www.ittig.cnr.it/loait/loait.html#top

OPJK into PROTON: Legal Domain Ontology Integration into an Upper-Level Ontology

Núria Casellas[1], Mercedes Blázquez[2], Atanas Kiryakov[3], Pompeu Casanovas[1], Marta Poblet[1], and Richard Benjamins[2]

[1] Institute of Law and Technology, Universitat Autònoma de Barcelona, Spain
{nuria.casellas, pompeu.casanovas, marta.poblet}@uab.es
[2] iSOCO, Intelligent Software Components, S.A., (Madrid) Spain
{mercedes,rbenjamins}@isoco.com
http://www.isoco.com
[3] Ontotext Lab, Sirma Group, (Sofia) Bulgaria
naso@sirma.bg
http://www.ontotext.bg

Abstract. The SEKT Project aims at developing and exploiting the knowledge technologies which underlie the Next Generation Knowledge Management, connecting complementary know-how of key European centers in three areas: Ontology Management Technology, Knowledge Discovery and Human Language Technology. This paper describes the development of PROTON, an upper-level ontology developed by Ontotext, and of the Ontology of Professional Judicial Knowledge (OPJK), modeled by a team of legal experts form the Institute of Law and Technology (IDT-UAB) for the *Iuriservice* prototype (a webbased intelligent FAQ for the Spanish judges on their first appointment designed by iSOCO). The paper focuses on the work done towards the integration of the OPJK built using a middle-out strategy into the system and top modules of PROTON, illustrating the flexibility of this independent upper-level ontology.

1 Introduction

SEKT (Semantically Enabled Knowledge Technologies) is a European research and development project within the European Commission's Sixth Framework Programme. This project aims at connecting complementary know-how of key European centers of excellence in Ontology and Metadata Technology, Knowledge Discovery and Human Language Technology, a leading commercial exponent of semantic web technology, together with a major European ICT organization.[1]

Ontologies play a central role within semantic web technologies and, for that reason, several domain ontologies are being modeled by the case study partners. This is the case of the Ontology of Professional Judicial Knowledge (OPJK). The OPJK ontology is developed within the SEKT Project by the Autonomous University of Barcelona (UAB) and iSOCO to support an intelligent FAQ prototype (*Iuriservice*).

[1] http://www.sekt-project.com

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 846 – 855, 2005.
© Springer-Verlag Berlin Heidelberg 2005

SEKT complementary know-how facilitates the integration of this domain ontology, the OPJK ontology, into PROTON, developed by Ontotext.[2] The PROTON ontology contains about 300 classes and 100 properties, providing coverage of the general concepts necessary for a wide range of tasks, including semantic annotation, indexing, and retrieval. The design principles can be summarized as follows (i) domain-independence; (ii) light-weight logical definitions; (iii) alignment with popular metadata standards; (iv) good coverage of named entities and concrete domains (i.e. people, organizations, locations, numbers, dates, addresses). The ontology is originally encoded in a fragment of OWL Lite, split into four modules: System, Top, Upper, and KM (Knowledge Management). This paper describes this integration and illustrates the flexibility of this independent upper-level ontology.

2 PROTON Development

PROTON (PROTo ONtology) ontology is developed by Ontotext Lab in the scope of the SEKT project as a light-weight upper-level ontology, which serves as a modeling basis for a number of tasks in different domains.

In a nutshell, PROTON is designed as a general-purpose domain-independent ontology. The above mission statement predetermines a couple of drawbacks (i) PROTON is (relatively) un-restrictive, and (ii) PROTON is naïve in many aspects, as for instance the conceptualization of space and time. Having accepted the above drawbacks, we put a couple of additional requirements towards PROTON, namely, to allow for (i) low cost of adoption and maintenance, and (ii) scalable reasoning. The high-level goal is to make feasible the usage of ontologies and the related reasoning infrastructure as a replacement for the DBMS.

2.1 Ontologies as RDBMS Schema

Here we discuss formal ontologies modeled through knowledge representation (KR) formalisms based on mathematical logic (ML); there is a note on the so-called topic-ontologies in a subsection below. If we compare the ontologies with the schemata of the relational DBMS, there is no doubt that the former represent (or allow for representations of) richer models of the world. There is also no doubt that the interpretation of data with respect to the fragments of ML, which are typically used in KR, is computationally much more expensive as compared to interpretation towards a model based on relational algebra. From this perspective, the usage of the lightest possible KR approach, i.e. the least expressive logical fragment, is critical for the applicability of ontologies in a bigger fraction of the contexts in which DBMS are used.

On our view, what is very important for the DBMS paradigm is that it allows for management of huge amounts of data in a predictable and verifiable fashion. This requirement is not well covered by the heavy-weight, fully-fledged, logically expressive knowledge engineering approaches. Even taking a trained knowledge engineer and a relatively simple logical fragment (e.g. OWL DL), it is too hard for the engineer to maintain and manage the ontology and the data, with the growth of the size of the

[2] http://proton.semanticweb.org/

ontology and the scale of the data. We leave the above statements without a proof, hoping that most of the readers share our observations and intuition.[3]

2.2 Formalization (Knowledge Representation) Approach

PROTON is originally formalized in OWL, more precisely, in a fragment of OWL Lite. The epistemological model of OWL is derived from the one of RDF. The world is modeled in terms of resources –everything is modeled as resource having a unique identifier (URI). The resources can "belong" or "be" instances of classes. The resources are described through triples of the form **<subject,redicate,object>**, which connect the resource (the subject), through a specific property (the predicate, the type of the relation) to the object, which could be either another resource, or a literal. The literals represent concrete data values (such as strings and numbers), essentially XML literals. Thus, a person can be described with a couple of statements as follows:

- **<#p1,type,#Person>** - determines the class of the resource;
- **<#p1,#hasFather,#p2>** - connecting the person to the resource representing her father;
- **<#p1,#hasBirthDate,"3/12/1973">** - connecting the person with the literal which represents her birth date.

PROTON defines a number of classes and properties. All classes which represent categories of objects or phenomena in the domain of discourse (generally, the world) are defined as sub-classes of **Entity**. We introduce the **Entity** class in order to distinguish the classes used to encode the proposed conceptualization from the others which have auxiliary and/or technical role in the RDF(S) vocabulary. Below we will briefly present the RDFS and OWL constructs which are used to define the classes and properties in PROTON, the list of all the modeling primitives used in PROTON, but just a fraction of the full vocabulary of OWL and even of OWL Lite:

- Both classes and properties have identifiers (URIs), titles (**rdf:label**) and descriptions (**rdf:comment**);
- Classes can be defined as sub-classes of other classes (via **rdf:subClassOf**). All classes are defined as instances of **owl:Class**, because in OWL the notion of class is more restrictive and clear as compared to RDF(S)).
- The properties are distinguished into object- and data-properties (respectively, **owl:ObjectProperties** and **owl:DataProperties**). The object-properties are referred to in some modeling paradigms as (binary) relations – they relate entities to other entities. The data-properties are often referred to as attributes – they relate entities to literals.

[3] We are tempted to share a hypothesis regarding the source of the unmanageability of any reasonably complex logical theory. It is our understanding that the Mathematical Logic is a coarse model of the human cognition – it provides a poor approximation for the process of human thinking, which renders it hard to follow. The relational algebra is also a poor approximation, but it seems simple enough, to be simulated by a trained person.

- Domain and ranges of properties are defined. The domain (**rdfs:domain**) specifies the classes of entities to which this property is applicable. The range (**rdfs:range**) specifies the classes of entities (for object-properties) or the datatypes (in case of data-properties).
- Properties can be defined as sub-properties of other properties (via **rdf:subPropertyOf**).
- Properties can be defined as symmetric and transitive (via **owl:SymmtericProperty** and **owl:TransitiveProperty**).
- Object-properties can be defined to be inverse to each other (via **owl:inverseOf**).

PROTON follows the principle of strict layering, which means that no classes or properties are instance of other classes.[4] We had limited ourselves to the above modeling means and principles for the following reasons: (i) this level of expressivity allows for straightforward reasoning and interpretation on the basis of standard extensional semantic; (ii) it matches the modeling paradigm of the object-oriented approach (the OO-databases, languages such as Java, etc.); (iii) it allows easy maintenance, integration and migration to other formalizations.

2.3 Topic-Ontologies, Taxonomies, and Subject Hierarchies

There is a wide range of applications for which the classification of different things with respect to hierarchy of topics, subjects, categories, or designators has proven to be a good organizational practice which allows for efficient management, indexing, storage, or retrieval. Probably the most classical examples in this direction are the library classification systems and also the taxonomies, widely used in the KM field. Finally, Yahoo and DMoz, are popular and very large scale incarnations of this approach in the context of the world wide web.

As long as the above mentioned conceptual hierarchies represent a form of a shared conceptualization, it is not a surprise that they are also considered a sort of ontologies. It is our understanding, however, that these ontologies bear a different sort of semantics. The formal framework, which allows for efficient interpretation of DB-schema-like ontologies (such as PROTON) is not that suitable and compatible with the semantics of the topic hierarchies. For the sake of clarity, we introduce the term "schema ontology" to refer to the first and the term "topic ontologies" to refer to the second sort. The schema-ontologies are typically formalized with respect to the so-called extensional semantics, which in its simplest form allows for a two-layered set-theoretic model of the meaning of the schema elements. It can be briefly characterized as follows:

- The set of classes and relations on one hand is disjoint from the set of individuals (instances), on the other (TBox and ABox vocabularies in the description logics).
- The semantic of the classes is defined through the sets of their instances. The subclass operation in this case is modeled as set inclusion (as in the classical algebraic set theory);

[4] The later (class as an instance of another class) is allowed by RDFS and OWL Full, but it changes the class of complexity of the logical fragment, which makes impossible the application of a range of efficient inference techniques.

- The relations are defined through the sets of ordered n-tuples (the sequences of parameters or arguments) for which they hold. Sub-relations, are again defined through sub-sets. In the case of RDF/OWL properties, which are binary relations, their semantic is defined as a sets of ordered pairs of subjects and objects;
- This model can easily be extended to provide mathematical grounding for various logical and KR operators and primitives, such as cardinality constraints.
- Everything which cannot be modeled through set inclusion, membership, or cardinality within this model is undistinguishable or "invisible" for this sort of semantics – it is not part of way in which the symbols are interpreted.

This sort of semantics can be efficiently supported (in terms of induction and deduction algorithms) to a well-known extent. It can also be extended in different directions – something that the logicians are doing for centuries. A typical and very interesting representative of this class are the description logics, and in particular OWL DL.

It is our understanding that the semantic of the topics has a different nature. Topics can hardly be modeled with set-theoretic operations – their semantic has more in common with the so-called intensional semantics. In essence, the distinction is that the semantic is not determined by the set of instances (the extension), but rather by the definition itself and more precisely the information content of the definition. The intensional semantic is in a way closer to associative thinking of the human being than the ML (in its simple incarnations) is. The criteria whether something is a sub-topic of something else have no much to do with the instances of the concrete class (if the topic is modeled this way). To some extent it is because the notion of instance is hard to define in this case.

Even disregarding the hypothesis for the different nature of the semantics of the topic-ontologies, we suggest that those should be kept detached from the schema-ontologies. The hierarchy of classes of the latter should not be mixed up with the topic hierarchies, because this can easily generate paradoxes and inconsistent ontologies. Imagine a schema-ontology, where we have definitions for **`Africa`** and **`AfricanLion`** – it is likely that **`Africa`** will be an instance of the **`Continent`** class and **`AfricanLion`** will be a sub-class of **`Lion`**. Imagine also a book classification – in this context **`AfricanLionSubject`** can be a subsumed by **`AfricaSubject`**. If we had tried to "re-use" for classification the definitions of **`Africa`** and **`AfricanLion`** from the schema-ontology, this would require that we define **`AfricanLion`** as a sub-class of **`Africa`**. The problems are obvious: one of this is not a class, and there is no easy way to redefine it so that the schema-ontology extensional sub-classing coincides with the relation required in the topic hierarchy.[5]

Based on the above arguments, we drew up a couple of principles and implemented those in PROTON: (i) the class hierarchy of the schema ontology should not be mixed with topic hierarchies; (ii) we should avoid precise and comprehensive modeling of

[5] This example was proposed by the one of the authors, to Natasha Noy for the sake of support of Approach 3 within the ontology modeling study published at http://smi-web.stanford.edu/people/noy/ClassesAsValues/ClassesAsValues-2nd-WD.html. One can find there some further analysis on the computational complexity implications of different approaches for modeling of topic hierarchies.

the semantics of topics within a computational environment based on extensional semantics.

The **`Topic`** class within PROTON is meant to serve as a bridge between two sorts of ontologies. The specific topics should be defined as instances of the Topic class (or a sub-class of it). The topic hierarchy is build using the **`subTopic`** property as a subsumption relation between the topics. The latter is defined to be transitive, but what is most important, it has nothing in common with the **`rdfs:subClassOf`** meta-property.

2.4 The Structure of PROTON

In order to meet the requirements of the usage scenarios and to assure easy and gradual understanding, PROTON is separated into four modules:

- **System module.** It contains a few meta-level primitives (5 classes and 5 properties). It introduces the notion of 'entity', which can have aliases. This module can be considered an application ontology. Within this document and in general, the System module of PROTON is referred to via the "**`protons:`**" prefix.
- **Top module.** The highest, most general, conceptual level, consisting of about 20 classes. These ensure a good balance of utility, domain independence, and ease of understanding and usage. The top layer is usually the best level to establish alignment to other ontologies and schemata. Within this document and in general, the Top module of PROTON is referred to via the "**`protont:`**" prefix.
- **Upper module.** Over 200 general classes of entities, which often appear in multiple domains. Within this document and in general, the Upper module of PROTON is referred to via the "**`protonu:`**" prefix.
- **KM (Knowledge Management) module.** 38 classes of slightly specalized entities that are specific for typical KM tasks and applications. The KM module is actually the former SKULO ontology, further developed and integrated into PROTON. Within this document and in general, the PROTON KM module is referred to via the "**`protonkm:`**" prefix.

The design at the highest level (immediately below the Entity class) of the Top module makes the most important philosophical and modeling distinctions – it follows the stratification principles of DOLCE [9],[6] through the establishment of the PROTON trichotomy of Objects (**`dolce:Endurant`**), Happenings (**`dolce:Perdurant`**), and Abstracts (**`dolce:Abstract`**).

A detailed documentation of the ontology, including its link to other ontologies and metadata standards can be found in [11] as well as at http://proton.semanticweb.org.

3 OPJK Development and Integration

The Ontology of Professional Judicial Knowledge (OPJK) has been built manually, extracting relevant concepts and relations from nearly 800 competency questions about practical problems faced by Spanish judges in their first appointment [2, 3, 4, 5].

[6] http://www.loa-cnr.it/DOLCE.html

The ontology will provide the semantic link between questions posed in natural language by the judges to an FAQ system and the relevant question-answer pairs stored within. Thus, this system will offer judges relevant answers through semantic matching. This prototype is called *Iuriservice* and is being developed by a team of legal experts from the Autonomous University of Barcelona (UAB) and iSOCO (Intelligent Software Components, S.A.) with the collaboration and supervision of the Spanish Council of the Judiciary.

It is important to note that the knowledge modeled corresponds to the professional (not the purely theoretical) knowledge contained within the judiciary and transmitted only between its members. Professional knowledge encodes a specific type of knowledge related to specific tasks, symbolisms and activities [1] and enables professionals to perform their work with quality [8]. Legal professions, especially the judiciary, not only share among themselves a portion of the legal knowledge constituted by legal language, statutes and previous judgments, but also knowledge related to personal behavior, practical rules, corporate beliefs, effect reckoning and perspective on similar cases, which remain tacit.

From this point of view, the design of legal ontologies requires not only the representation of normative language but also the representation of the professional knowledge derived from the daily practice at courts, a knowledge that is not being captured by the current trends in legal ontology modeling. Modeling this professional judicial knowledge demands the description of this knowledge as it is perceived by the judge and the attunement of dogmatic legal categorizations; the assumption that their reasoning process follow some specific dogmatic patterns is not required.[7]

Capturing this professional knowledge is a time consuming and meticulous process that requires the use of different sociological techniques (field work, etc.) and the gathering of empirical data, which will influence ontological modeling, as learning ontologies from this data "situates" the knowledge. Thus, it is necessary that the methodology followed during the construction of this ontology focuses on the maintenance of this point of view. For this reason, OPJK has been developed following the middle-out strategy [10] that consists on the specification or generalization, when needed, of the identified terms and its integration into a top ontology has to take into account all this "situated knowledge".

The OPJK methodology construction has been specifically described and presented in a number of papers and conferences [6]. In this paper, we focus in the integration of OPJK into PROTON, the top ontology described above for the purposes of providing an upper conceptual framework. This integration not only allows the reuse of an existing upper ontology, PROTON but also allows the maintenance of the necessary "professional" trait of OPJK. This integration has taken place at two stages: OPJK concept into PROTON concept integration and the reuse of existing PROTON relations.

3.1 Class Integration

The first part of the integration process consisted mainly in generalizing OPJK concepts taking into account the System and Top modules of PROTON, fully

[7] We use *situated knowledge* in a similar way to Clancey's [7] "situated cognition": "an approach for understanding cognition that seeks to relate social, neural, and psychological views."

incorporating the meta-level primitives contained in the System module (*i.e.* "**Entity**") as the application ontology.

As stated above, the top layer of PROTON is the best level to establish the alignment with OPJK. It has proved to be domain independent and its easy understanding and usage have been essential to guarantee this integration. The primary classes **Abstract, Happening** and **Object** were straightforwardly incorporated, although **Abstract** needed the introduction of a specific subclass **AbstracciónLegal** [LegalAbstraction] for organizational purposes. With this domain specific consideration, the OPJK classes **CalificaciónJurídica** [LegalType] **Jurisdicción** [Jurisdiction] and **Sanción** [Sanction] could be better related and specific relations between them, not shared by the rest of classes/instances within **Abstract.**

The class of entities **Happening**, which includes **Event, Situation and TimeInterval** is able to directly incorporate the fundamental OPJK classes **Acto** [Act] (**ActoJurídico** [JudicialAct]), **Fase** [Phase] and **Proceso** [Process]. These classes contain the taxonomies and relations related to all legal acts, to the different types of judicial procedures (subdivided within civil and criminal judicial procedures) and the different stages that these procedures can consist in (period of proof, conclusions,…). A necessary reference has to be made to the introduction in PROTON of the class **Role**, which allowed the distinction of situations where an agent (**Organization** or **Person**) might play a part in situations that differ from the more business-related **JobPosition** [11]. In the case of OPJK, the class **Role** contains the concepts and instances of procedural roles (**RolProcesal**) that an agent might play during a judicial procedure.

Finally, **Object** includes the top OPJK classes **Agent** and **Statement** that are generalizations for **Document**, and **Location**, necessary concepts to contain, within others, the organizational taxonomy of courts (**OrganoJudicial**), and judicial documents (**Contrato** [Contract], **Recurso** [Appeal], **Sentencia** [Judgment], etc.).

3.2 Inherited Relations

The specificity of the legal (professional) domain requires specific relations between concepts (normally domain-related concepts as well). However, most existing relations between the Top module classes taken from PROTON have been inherited and incorporated. It has not been necessary for the usage of the *Iuriservice* protoype to inherit all PROTON relations, although most of the relations contained in PROTON had already been identified as relations between OPJK concepts.

The following relations –not a comprehensive list– have been inherited from the existing relations in within the Top module concepts: **Entity hasLocation, Happening** has **endTime** and **startTime, Agent** is **involvedIn** (**Happening**), **Group hasMember,** an **Organization** has **parent/childOrganizationOf** (**Organization**) and is **establishedIn,** and, finally, **Statement** is **statedBy** (**Agent**), **validFrom** and **validUntil**.

4 Conclusions

The work described in this paper refers to the integration of the ontology of professional judicial knowledge (OPJK) built using a middle-out strategy into an upper level ontology. The SEKT complementary know-how facilitates the integration of this domain ontology, the OPJK ontology, into the System and Top modules of PROTON, developed by Ontotext.

This paper confirms that the Top module of PROTON ensures a good balance of utility and ease of usage and understanding. As top layers represent usually the best level to establish alignment to other ontologies, the classes contained in the Top module **`Abstract, Happening`** and **`Object`** were straightforwardly incorporated, together with most of their subclasses. In the same way, most of the existing relations/properties between the Top module classes were also inherited.

The essential domain independence of PROTON has fostered this integration and both, the reuse of existing PROTON classes and relations and the creation of new specific legal domain subclasses or relations to facilitate OPJK integration illustrates the flexibility of this domain independent top ontology.

Acknowledgements

SEC2001-2581-C02-01 and EU-IST (IP) IST-2003-506826 SEKT.

References

1. Abel, R.L. (Ed.) (1997). *Lawyers: A Critical Reader*. The New Press, New York.
2. Benjamins, V.R; Contreras, J.; Casanovas, P.; Ayuso, M.; Bécue, M.; Lemus, L.; Urios, C. (2004a). "Ontologies of Professional Legal Knowledge as the Basis for Intelligent IT Support for Judges" *Artificial Intelligence and Law*, in press.
3. Benjamins, V,R.; Contreras, J.; Blázquez, M.; Rodrigo, L.; Casanovas, P.; Poblet, M. (2004b). "The SEKT use legal case components: ontology and architecture", in T.B. Gordon (Ed.), *Legal Knowledge and Information Systems*. Jurix 2004. IOS Press, Amsterdam, 2004, pp. 69-77.
4. Benjamins V.R.; Casanovas, P. Contreras, J., López-Cobo, J.M.; Lemus, L. (2005). "Iuriservice: An Intelligent Frequently Asked Questions System to Assist Newly Appointed Judges", in V.R. Benjamins et al. *Law and the Semantic Web*, Springer-Verlag, London, Berlin, pp. 205-22.
5. Casanovas, P., Poblet, M., Casellas, N., Vallbé, J.J., Ramos, F., et al (2004). *D 10. 2. 1 Legal Scenario* (WP10 Case Study: Intelligent integrated decision support for legal professionals report EU-IST Integrated Project (IP) IST-2003-506826 SEKT), 2004.
6. Casanovas, P. Casellas, N., Tempich, C., Vrandečić, D., Benjamins, R. (2005). "OPJK modeling methodology", Lehmann, J., Biasiotti, M.A., Francesconi, E., Sagri, M.T. (eds.) (2005). *LOAIT –Legal Ontologies and Artificial Intelligent Techniques– IAAIL Workshop Series,* Wolf Legal Publishers, Nijmegen, The Netherlands, pp. 121-133.
7. Clancey, W.J. ; Sachs, P. ; Sierhus, M.; Hoof, R.v. (1998). "Brahms. Simulating practice for work systems design", *International Journal of Human-Computer Studies* no. 49, pp. 831-865.

8. Eraut, M. (1992). "Developing the knowledge base: a process perspective on professional education". In. Barnett, R. (Ed.) *Learning to effect*. Open University Press, Buckingham, pp. 98-118.
9. Gangemi, A.,Guarino, N., Masolo, C., Oltramari, A., Schneider, L. (2002). *Sweetening Ontologies with DOLCE.* In Proc. of 13th Int. Conference on Knowledge Engineering and Knowledge Management EKAW02, Sigüenza, Spain, 1-4 October 2002.
10. Gómez-Pérez, A.; Corcho, O.; Fernández-López, M. (2002). *Ontological Engineering*, Springer-Verlag, London, Berlin.
11. Terziev, I., Kiryakov, A., Manov, D. (2004) *D 1.8.1. Base upper-level ontology (BULO) Guidance,* report EU-IST Integrated Project (IP) IST-2003-506826 SEKT), 2004. http://proton.semanticweb.org/D1_8_1.pdf

Semantic Transformation of Web Services

David Bell, Sergio de Cesare, and Mark Lycett

Brunel University,
Uxbridge, Middlesex,
UB8 3PH, United Kingdom
{david.bell, sergio.decesare, mark.lycett}@brunel.ac.uk

Abstract. Web services have become the predominant paradigm for the development of distributed software systems. Web services provide the means to modularize software in a way that functionality can be described, discovered and deployed in a platform independent manner over a network (e.g., intranets, extranets and the Internet). The representation of web services by current industrial practice is predominantly syntactic in nature lacking the fundamental semantic underpinnings required to fulfill the goals of the emerging Semantic Web. This paper proposes a framework aimed at (1) modeling the semantics of syntactically defined web services through a process of interpretation, (2) scoping the derived concepts within domain ontologies, and (3) harmonizing the semantic web services with the domain ontologies. The framework was validated through its application to web services developed for a large financial system. The worked example presented in this paper is extracted from the semantic modeling of these financial web services.

1 Introduction

Web services have become the predominant paradigm for the development of distributed software systems. Web services provide the means to modularize software in a way that functionality can be described, discovered and invoked in a platform independent manner over a network (e.g., intranets, extranets and the Internet). Notwithstanding the architectural advantages of such a paradigm, the representation of web services by current industrial practice is predominantly syntactic in nature lacking the fundamental semantic underpinnings required to fulfill the goals of the emerging Semantic Web.

Within a Semantic Web context web services require precise semantic representations, normally achieved through the use of ontologies, in order to provide the necessary relationships with domain models and ultimately mappings to the real world objects that such models refer to. As a consequence, syntactic web services already described in languages like the Web Services Description Language (WSDL) require semantic transformations and subsequent integration with domain ontologies [1].

The de facto standard languages for describing, publishing and invoking web services are currently WSDL, Universal Description, Discovery and Integration (UDDI), and Simple Object Access Protocol (SOAP), respectively. Although such languages provide the technical means for achieving cross-platform distributed software deployment, they are not sufficient to achieve a level of semantic expression necessary

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 856 – 865, 2005.
© Springer-Verlag Berlin Heidelberg 2005

for machines to automatically relate web services to other resources and in doing so discover the services required for composing and choreographing the intended behavior [2]. In relation to the Semantic Web and its goals, syntactically defined web services represent legacy applications which need to be conceptually reengineered in order to extract the semantics (i.e., precise meaning) of the intended behavior and the underlying domain concepts such behavior utilizes. This conceptual reengineering can be referred to as semantic transformation. The ultimate result of semantic transformation is a set of ontological models which would allow an agent (typically a software agent) to navigate through a semantic network which contains references to all types of web resources including available services.

This paper presents a framework aimed at (1) modeling the semantics of syntactically defined web services through a process of interpretation, (2) scoping the derived concepts within domain ontologies, and (3) harmonizing the semantic web services with the domain ontologies. The framework was validated through its application to web services developed for a large financial system. The worked example presented in this paper is extracted from the semantic modeling of these financial web services.

2 Lack of Semantics in Web Services

Web services are a fundamental part of the emerging Semantic Web. Web services are self-contained and self-describing modular Web applications that can be published, located, and invoked across a network (IBM) and capable of supporting interoperable machine-to-machine interaction (World-Wide Web Consortium). A web service has an interface described in a machine-processable format. It is through this interface that a web service communicates with other software applications. Although the tools and methods required to develop web services have matured over recent years, there exists limited support in the area of the semantic representation of web services and their integration with other web resources [3]. Such a need is motivated by the machine-processable nature of all Semantic Web resources. In order for web services to be automatically discovered, selected and composed, it is necessary for a software agent to autonomously navigate the Semantic Web in search of services satisfying specific criteria. Such criteria are generally defined in terms of what a service provides (i.e., output) and what a service requires (i.e., input). For a software agent to recognize such elements, both inputs and outputs should preferably be expressed or typed with reference to ontological models which semantically map to the resources (including domain objects and web services) of the Semantic Web. With such models all web resources would be represented through interrelated web ontologies, thus facilitating the integration of web services with the whole of the Semantic Web.

Currently the scenario just described is not implemented. Web services are primarily adopted in industry as a means to develop architecturally sound information systems. Web services as they are typically developed today do not support the necessary semantic precision and "machine-processability" for software agents to automatically navigate through the future Semantic Web and pinpoint those services which can suit specific requirements. As this research shows, web services developed in industry today are mostly syntactic in nature. This is simply demonstrated by the elementary typing of the services' input and output parameters. Such parameters are normally

typed in relation to traditional programming language types such as strings, integers and floats. Such types do not map directly to web resources such as books, flights, bank accounts and people. As such a software agent searching for a flight booking service would unlikely find a service with an input parameter typed by an ontologically represented 'flight' class, but would most probably find many services with generic string input parameters. Such syntactic representations work in a semantically poor environment based on WSDL, UDDI and SOAP, however they would not be able to scale up to the requirements of the Semantic Web as described above.

More recently serious and important attempts have been undertaken to define languages for semantically representing web services. Initiatives such as OWL-S and WSDL-S are an important step forward. OWL-S, for example, defines a high level ontology for web services. OWL-S is based on OWL (Web Ontology Language) and as such provides the basis for the possible semantic integration between web services and other web resources. In the presence, however, of a vast amount of syntactic web services developed to date, service ontologies like OWL-S are necessary but not sufficient to resolve the problem of how to integrate these technical services within the emerging Semantic Web. In addition, much of the current research assumes the existence of ontology for composition or discovery [4]. A framework for systematically transforming syntactic web services into semantic web services is required to support these assumptions. The remainder of this paper will present such a framework and exemplify it within the context of a financial services example.

3 Framework

3.1 Underlying Philosophy and Concepts

A framework has been developed for deriving semantic content from syntactic web services and representing such semantics in ontological models. The framework is based on the principles of content sophistication described by Partridge [5] and Daga et al. [6]. Content sophistication represents a process for improving the semantic contents of legacy systems along several dimensions and representing such improvements in technology-agnostic conceptual models. The framework proposed in this paper provides the basis for interpreting the semantics of syntactic web services in a similar fashion. In fact in order to achieve the claimed benefits of the Semantic Web, it is necessary for web services to be semantically well defined and related to other types of web resources [7]. In this sense it is not exaggerated to state that, for the Semantic Web, syntactic descriptions of services developed today represent the 'legacy of the future'.

At the heart of the framework is the adoption of ontology to drive the derivation of semantic content from syntactic web services. From a philosophical perspective ontology can be defined as a set of things whose existence is acknowledged by a particular theory or system [8]. Such 'things' include both types (such as the class of *Bank Accounts*) and individual elements (such as *John Smith's Bank Account*). The adoption of such a definition is important because, when compared with more computationally orientated definitions of ontology (for example, Gruber [9] states that "an ontology is a specification of a conceptualization"), there is an explicit reference to a system's ontic commitment (i.e., things whose existence is acknowledged or

recognized). This leads to representations that are more closely mapped to real world objects. Such mapping or reference [10] is essential to ontological modeling. The meaning of a sign, used, for example, to denote a service or a parameter, becomes well understood when it is possible to identify the thing(s) the sign refers to.

The focus of the framework presented in this section is the discovery of the semantics underlying a service description in its fundamental parts (mainly name and parameters). This process of concept discovery, called interpretation, identifies those real world objects that individual service parts ontologically commit to (or refer to). The semantics that are unraveled in this way are then represented in technology-agnostic domain and service ontology models.

The framework addresses the following objectives: (1) Derivation of semantics from previously developed web service syntactic descriptions; (2) Representation of the derived semantics in ontological models; and (3) Integration of models of semantic web services with models of other web resources. These objectives define the scope of the framework. A process was defined in order to achieve the objectives listed above. It is beyond the scope of this paper to describe in detail how the ontological models derived from the framework can be used by a semantic web search facility to discover and compose services.

3.2 Framework Process and Artifacts

The process, which drives the discovery and representation of semantic content from technical web services, is summarized in Table 1. The process is iterative and its outcome (defined in terms of ontological models) outlives one specific reengineering project. The framework's ongoing mission is to develop (within and across domains) interlinked ontological models for the Semantic Web. These models represent simultaneously all types of resources including service offerings. The process consists of three main activities: service interpretation, concept scoping and harmonization.

Table 1. Process for deriving semantic content from web services

Activities	Description	Input Artifacts	Output Artifacts
Service interpretation	A service description is broken down into its fundamental parts (e.g., name, input and output parameters). Each part is interpreted in order to represent its ontic commitment.	▪ Web service descriptions (e.g., WSDL code)	▪ Individual service ontic commitment models
Concept scoping	The concepts represented in the service ontic commitment models are either mapped to pre-existing ontologies or assigned to newly developed ones.	▪ Service ontic commitment models ▪ Domain ontologies	▪ Objects incorporated or mapped to ontological domain models
Harmonization	Services are represented within ontological models and related to other domain objects.	▪ Service ontic commitment models ▪ Domain ontologies	▪ Extended or specialized domain ontology ▪ Service ontology

These activities have been adapted from the Content Sophistication process presented by Daga et al. [6]. As a whole the process takes in technical service descriptions and produces ontological representations. The individual process activities also require and produce artifacts which progressively lead to achieving the ontological models.

3.3 Interpretation

The first activity is Service Interpretation. This activity works on service descriptions with limited or no explicit semantic underpinning. The descriptions are normally represented in the form of a service name with input and output parameters. The parameters themselves are named and typed. For example, in WSDL a typical service description can be found as a combination of service signatures and data type definitions.

Interpretation is aimed at representing the service's ontic commitment. This means unbundling and making as explicit as possible the real world (business) objects that the service descriptions recognize the existence of. In fact interpretation is defined as "the act of clarifying or explaining the meaning" of something (Collins Concise Dictionary 2001, p.761). Analogously identifying the real world objects that a service commits to is an act of clarifying the meaning of service descriptions.

Interpretation produces Service Ontic Commitment (SOC) models adopting the Object paradigm [6]. The Object paradigm, not to be confused with the Object-Oriented paradigm, was specifically designed for business modeling and is quite effective in precisely representing real-world semantics. Precise representation, in this case, refers to being able to clearly identify the mappings between the representation and the represented. It is beyond the scope of this paper to describe the Object paradigm in detail. It is sufficient to note that this paradigm models all "things" (including classes, individuals and relationships) as objects with a four-dimensional extension. The paradigm is attribute-less unlike more traditional paradigms (e.g., entity-relationship or object-oriented).

3.4 Concept Scoping

Concept Scoping is aimed at allocating the "committed" objects of the SOC models to pre-existing ontological models or, in the case of a newly explored domain, to newly developed ontologies. There are various ways in which content scoping can occur. With reference to an ontology language like OWL new objects (such as classes, properties and individuals) can be incorporated into an ontology as exemplified in Table 2.

Table 2. Methods of incorporating identified classes, properties and individuals

Object Type	Method of Incorporation
Class	Define the class (a) in a newly developed ontology without any relation to pre-existing ontologies, (b) as a subclass of a class defined in a pre-existing ontology, (c) as an instance of a class defined in a pre-existing ontology and (d) as equivalent to a pre-existing class
Property types	Same as for classes
Individuals	Instantiate a class

3.5 Harmonization

Web services are resources which provide agents (human or software) with business offerings whose instantiations produce real world effects. Web services can use other web resources and can produce new resources. In this sense services will become an integral part of the Semantic Web and as such should be modeled similarly and in relation to other types of web resources. Harmonization is aimed at overcoming the traditional divide that is generally adopted between static and dynamic resources. The argument here is that if distinct types of representations are used for web services and other resource types, the necessary integration and semantic binding between them would become more difficult to resolve. Ontological models, which simultaneously represent all types of web resources, provide the benefit of facilitating the semantic discovery and composition of web services by software agents [11]. Agents would be able to traverse semantic graph lattices (or networks) in which services would be associated with the objects they use, transform and produce.

Harmonization uses the SOC models produced by Service Interpretation and the domain ontologies used in Concept Scoping to produce domain ontologies which incorporate service representations. The output artifact is represented in an ontology language such as OWL.

4 Financial Services Case

The research is grounded in a financial services case study which provides: (1) An external validity to the data that is seeding both the framework design process and subsequent scenario based usage analyses; (2) Less bias in that the software services being analyzed are the result of a service-orientation plan that did not encompass semantic web motivations; (3) A likely future industrial application of semantic web technology as tools and techniques mature and are accepted within such a commercial context.

M-Bank is a leading European bank with both retail and treasury banking operations. The case being investigated resides in the treasury operation. Web services are used to support the reuse of functionality within a core processing system. This functionality comprises the management of trade cash flows and the rate fixing process. Trades may live for up to several decades and involve the transfer of cash between the two contracted parties involved in the trade (monthly, quarterly, etc.). Over time, fixing rates are applied to trades allowing the resulting cash flow to be calculated. It is the fixing rates and the cash flow schedules that are of interest to the trader, as these changes have both a funding and hedging impact. Web services were used to allow the spreadsheet trading console to interact with the operational system that holds the cash flows and fixing rates.

5 Semantic Transformation Applied

The framework presented in this paper was applied to web services described in WSDL. The WSDL code specified about 50 operations with relative parameters. Each operation provides externally accessible offerings and as such can be considered web

services in their own right. The worked example presented in this section represents an extract of the semantic transformation carried out. This example refers to two web services. The first getRateSet returns the interest rate that has been fixed for a given trade settlement. The second web service is called getSchedule and returns the schedule of actual and projected settlements at a given point in time. The service receives as input reference to the trade and provides as output a table comprising of start and end dates of settlements, date in which the interest rate will be fixed (decided) for a specific settlement, currency, fixed rate, the notional amount of the settlement and the actual interest. From a semantic perspective such a representation has high levels of implicit meaning which need to be extracted and explicitly modeled. Tables 3 and 4 summarize the services.

Table 3. *getrateset* web service

Service name: getRateSet	
Description	This service provides the interest rate that has been fixed for a given settlement.
Input parameters	getRateSetSoapIn: String
Output parameters	getRateSetSoapOut: String

Table 4. *getSchedule* web service

Service name: getSchedule	
Description	This service provides the schedule of all settlements related to a given trade.
Input parameters	getScheduleSetSoapIn: String
Output parameters	getScheduleSetSoapOut: String

5.1 Interpretation

As the diagrams of Figures 1 show, each part of a service can be unbundled and mapped to real world objects that clearly define the part's semantics. Figure 1 specifically refers to the *getRateSet* web service. Additionally for the interpretation of *getSchedule*, the service name refers to the classes *Trades* and *Schedules*, and the

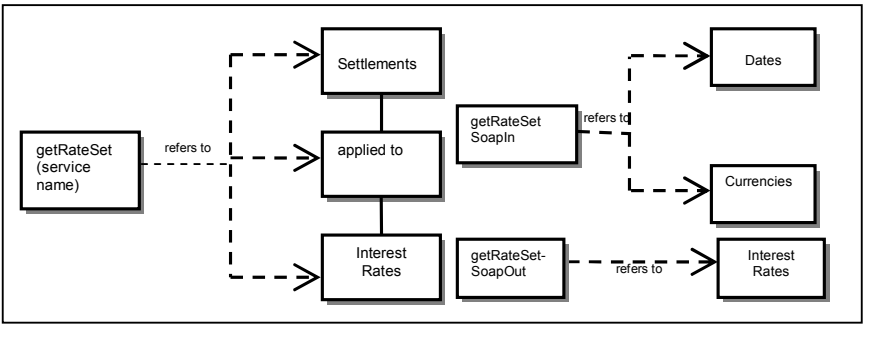


Fig. 1. Interpretation of *getRateSet* service name, input and output parameters

temporally organize relationship. Its input parameter, *getScheduleSoapIn*, relates to Trades and its output parameter, *getScheduleSoapOut*, relates to *Start Dates*, *End Dates*, *Rate Fixing Dates*, *Currencies*, *Interest Rates*, *Notional Amounts* and *Interest*.

The object paradigm, as stated previously, helps in this unbundling process given that all objects are explicitly revealed. The Service Ontic Commitment models shown here explicitly highlight those objects (in this case classes) that the individual elements of the services recognize the existence of, hence referring to such objects. Even relationships, such as *applied to* are represented as "committed" objects. This type of representation is similar to OWL in which relationships are explicitly represented as properties.

5.2 Content Scoping

Content scoping allocates the objects identified in the Service Ontic Commitment models to domain ontologies. Within this example it is assumed that a decision was taken to develop a financial ontology and to allocate all objects to such a model. However classes such as D*ates*, S*tart Dates* and *End dates* are typical candidates of classes that are most likely to be scoped within the context of previously existing ontologies. In this case a *Time* ontology would most probably contain the definition of a *Date* class. As such it would be advisable to refer to such a class and subtype it with classes such as *Start Dates* and *End Dates*.

Figure 2 illustrates a first-cut ontological model derived from the previous interpretation phase.

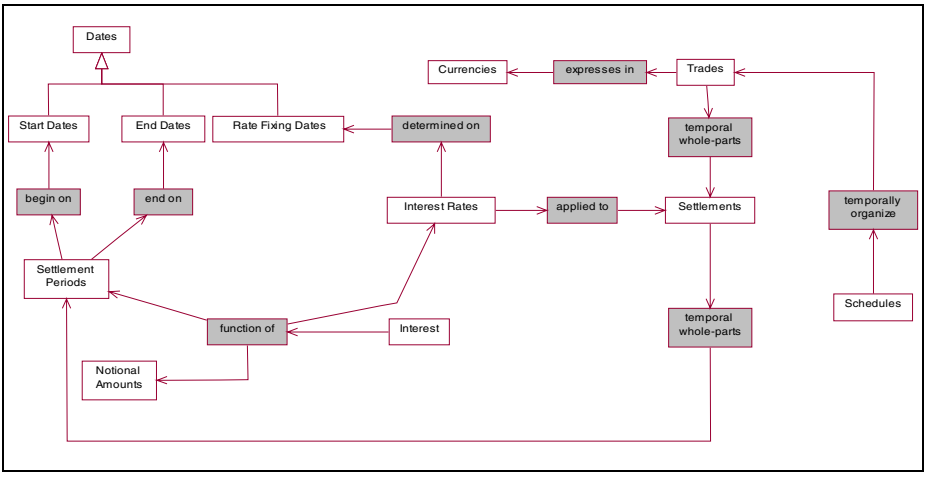


Fig. 2. First-cut financial domain ontology

5.3 Harmonization

In harmonization the web services are combined with the domain ontology. Ontologically this enables an explicit mapping between a service (with its parts) and the

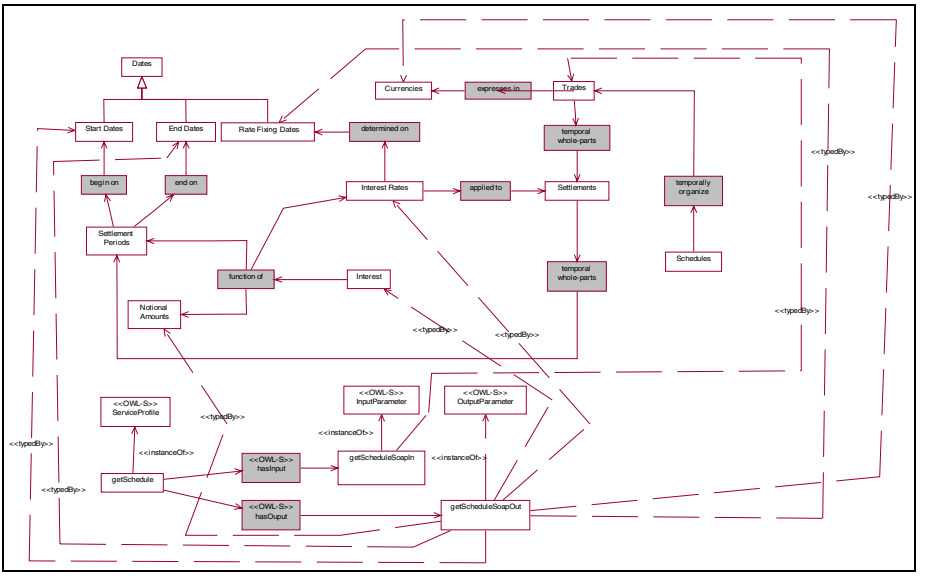


Fig. 3. Harmonized model

domain it serves. Figure 3 illustrates the harmonization model derived from the previous interpretation and content scoping.

6 Conclusion

This paper presented a framework for enabling the semantic transformation of syntactically defined web services. The framework defines an ontologically-based process in which syntactic service descriptions are interpreted to derive the objects that the services ontologically commit and refer to. The models produced by the interpretation phase are then used to scope the objects identified. These objects are either scoped to pre-existing web domain ontologies or used to develop new ontologies. Finally, the web services themselves are integrated with the domain ontologies. The final integration provides the basis for an effective semantic merging between all types of web resources and, as a consequence, facilitate the task of a software agent to navigate among various and semantically interlinked web services and domain objects (such as books, flights, etc.).

References

1. K. Sycara, M. Paolucci, J. Soudry, and N. Srinivasan, "Dynamic discovery and coordination of agent-based semantic Web services," *Internet Computing, IEEE*, vol. 8, pp. 66-73, 2004.
2. A. Paar, "Semantic software engineering tools " in *Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications* Anaheim, CA, USA ACM Press, 2003 pp. 90-91

3. S. Staab, W. van der Aalst, V. R. Benjamins, A. Sheth, J. A. Miller, C. Bussler, A. Maedche, D. Fensel, and D. Gannon, "Web services: been there, done that?," *Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications]*, vol. 18, pp. 72-85, 2003.
4. S.A. McIllraith, D.L. Martin , "Bringing Semantics to Web Services" *Intelligent Systems, IEEE*, vol. 18, pp. 90-93, 2003.
5. C. Partridge, *Business Objects: Re-Engineering for Reuse*. Oxford: Butterworth-Heinemann, 1996.
6. A. Daga, S. de Cesare, M. Lycett, and C. Partridge, "An Ontological Approach for Recovering Legacy Business Content," in *Proceedings of the 38th Hawaii International Conference on System Sciences*. Los Alamitos, CA: IEEE Computer Society, 2005.
7. D. Fensel and O. Lassila, "The semantic web and its languages," *Intelligent Systems and Their Applications, IEEE [see also IEEE Intelligent Systems]*, vol. 15, pp. 67-73, 2000.
8. T. Honderich, *Oxford Companion to Philosophy*. Oxford: Oxford University Press, 1995.
9. T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, pp. 199-220, 1993.
10. G. Frege, *The foundation of Arithmetic: A logico-mathematical enquiry into the concept of number*, 1884.
11. J. Hendler, "Agents and the Semantic Web," *Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications]*, vol. 16, pp. 30-37, 2001.

Modeling Multi-party Web-Based Business Collaborations*

Lai Xu and Sjaak Brinkkemper

Institute of Information and Computing Sciences,
Utrecht University, 3584CH Utrecht, The Netherlands
{L.Xu, S.Brinkkemper}@cs.uu.nl

Abstract. To remain competitive, enterprises have to mesh their business processes with their customers, suppliers and business partners. Increasing collaboration includes not only a global multi-national enterprise, but also an organization with its relationship to and business processes with its business partners. Standards and technologies permit business partners to exchange information, collaboration and carry out business transaction in a pervasive Web environment. There is however still very limited research activity on modeling multi-party Web-based business collaboration underlying semantics. In this paper, we demonstrate that an in-house business process has been gradually outsourced to third-parties and analyze how task delegations cause commitments between multiple business parties. Finally we provide process semantics for modeling multi-party Web-based collaborations.

1 Introduction

In the modern business world, we see that explicit structural collaboration between organizations is becoming more and more important. This is reflected in the emergence of tightly coupled supply chains, the service outsourcing paradigm, complex co-makerships, etc. Collaboration is not limited by geographical proximity, but increasingly of an international character. As a result of this, explicit multi-party business coordinations are becoming global. The need for a multi-party collaboration model for a business process is thus becoming evident.

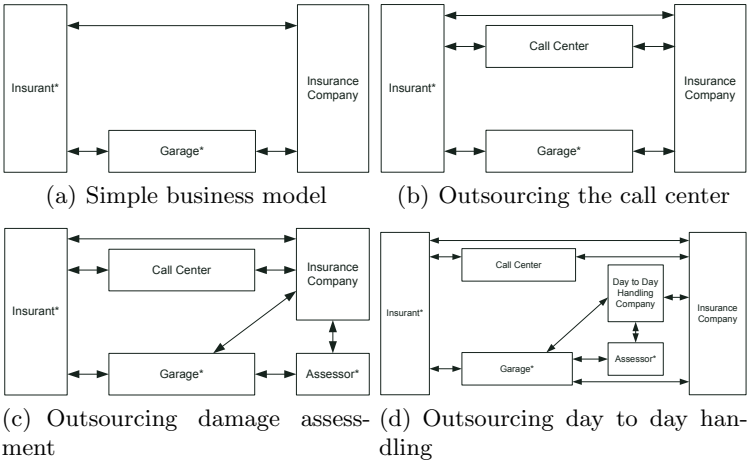
In the rest of this paper, we first elaborate how an in-house business process has been gradually outsourced in Section 2. In Section 3, we define our modeling language for multi-party business collaborations. We evaluate relate work in this area in Section4. The paper concludes with an summary and directions for further research in Section 5.

2 Multi-party Business Collaborations and Outsource

We provide a car insurance case for explaining how a car insurance business is gradually outsourced and in which collaborations are involved afterward. At the

* This work is sponsored by the Netherlands national research organization NWO, Jacquard project Overture (project number: 638.001.203).

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 866–875, 2005.
© Springer-Verlag Berlin Heidelberg 2005



(a) Simple business model (b) Outsourcing the call center

(c) Outsourcing damage assessment (d) Outsourcing day to day handling

Fig. 1. Car Insurance Business Processes

start time, a car insurance company probably only involves a group of garages to assess car damages and to repair damaged cars for an insurant, who has bought car insurance from the car insurance company. The insurance company deals with the rest of the issues. More precisely, after the occurrence of a car damage, a process starts, including many interactions among the insurant, a garage and the insurance company (see Figure 1 (a)).

After some time, the insurance company decides to outsource the phone service to a call center. The business process is consequently changed (along the line of Figure 1 (b)). The call center is responsible for registering the insurant information, suggesting an appropriate garage (most time a close by garage is assigned) and notifying the insurance company about the insurant's claim. Except the phone service, the insurance company still needs to handle the rest of services for the insurant.

Continuing it could be an alternative to outsource the inspection of damaged vehicles to an association of assessors. In this business model (see Figure 1 (c)), the assessors conduct the physical inspections of damaged vehicles and agree upon repair figures with the garages. After the call center, the garages and the assessors finish their obligations, the insurance company performs the rest services.

Due to the increasing amount of insurants, the insurance company might finally decide to outsource the daily service to a day to day handling company. The day to day handling company coordinates and manages the operation on a day-to-day level on behalf of the insurance company(see Figure 1 (d)). The detailed obligations of the day to day handling company are provided as follows. After receiving the forward claim from the insurance company, the day to day handling company will agree upon repair costs if an assessor is not required for small damages; otherwise, an assessor will be assigned. After finishing repairs, the garage will issue an invoice to the day to day handling company, which in

turn will check the invoice against the original estimate. The day to day handling company returns all invoices to the insurance company monthly. As a result the workload of the insurance company is significantly reduced.

Changes of the business models do not necessarily go through from Figure 1(a) to (b), then from (b) to (c) and finally from (c) to (d). Changes can happen, for example, directly from (a) to (d). Figure 1 demonstrates that how a business process is collaborated by more business parties in different circumstances. It also shows some essential characters of multi-party collaborations. One of them is that it is critical to understand *when* and *who* did, is doing or will do *what* in a multiple parties involved business process.

3 Multi-party Collaboration Modeling Language

In the business domain, we need to provide detailed and precise descriptions of multi-party business collaborations. In order to represent construct in the business domain, a language for modeling multi-party collaborations should be sufficiently expressive to represent a multi-party collaboration:

- in terms of its structure: who are the parties involved, and how are they interconnected,
- in terms of the commitments associated with those parties,
- and in terms of its processes: what actions are performed by which parties after which properties are satisfied.

An overview of the basic modeling concepts and their relationships is given as a metamodel in Figure 2. A multi-party collaboration consists of roles, parties, channels, commitment, actions and parameters. Parties perform different roles, fulfill different commitments and have many parameters. The roles perform actions. A channel connects two or more parties. A commitment aggregates many actions.

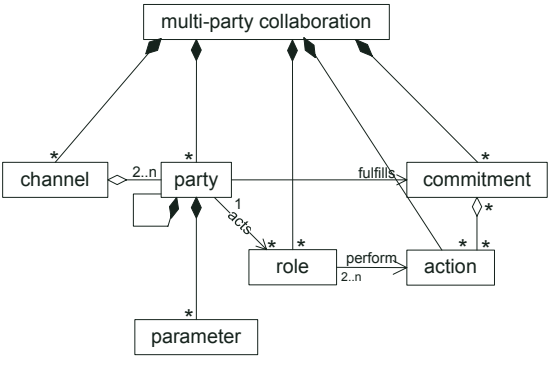


Fig. 2. Metamodel for Multi-party Collaboration Model

3.1 Collaboration Structure Model

Collaborated parties involved in a business process are modeled in collaboration structure model. Depending on the chosen scope of the model, *parties* may represent individual people, organizational units, such as departments, or an entire organization. Furthermore, a party can perform certain commitments.

Parties interact via *channels*, through which they may exchange information, goods or money. Channels are characterized by a medium (such as Internet, public switched telephone network (PSTN)), by transport (e.g. post, shipping or other ways). Figure 3 shows business parties collaboration model. It depicts a part of the business collaboration consisting of the insurance company and its co-operators.

Modeling collaboration structure is useful to identify the parties involved in business collaboration. It also provides a further step to clarify the responsibilities of parties.

3.2 Modeling Commitments Between Parties

To model commitments between multi-parties, we provide speech act theory and its extension and define commitments and a commitment model respectively.

Determining the Responsibilities of Parties. Part of Austin's work on speech act theory [1], is the observation that utterances are not implied propositions that are true or false, but attempts on the part of the speaker that succeed or fail. Perfomatives, acts, or actions are organized as speech acts and non-speech acts. An individual speech act is either a *solicit*, which explains an attempt to achieve mutual belief with the addressee that the sender wants the addressee to perform an act relative to the sender's wanting it done, or an *assert*, which expresses an attempt to achieve mutual belief with the addressee that the asserted statement is true.

The model of speech acts and repartee developed by Longacre recognizes two kinds of relations among successive utterances: replay and resolution in [2] and another two kinds of relations: resolves and completes by Van Dyke Parunakin in [3]. In the model of speech acts and repartee, every utterance in a conversation except for the first must "respond", "reply", "resolve" or "complete" to

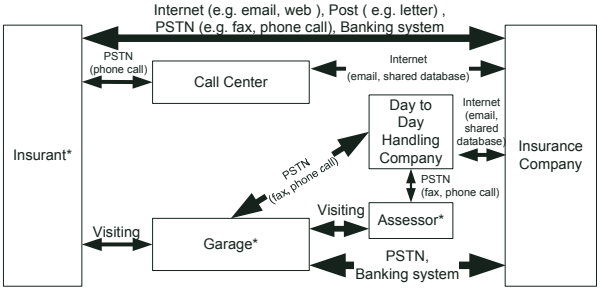


Fig. 3. Business Parties Related by Channels

another, otherwise there would be no conversation. Analyzing relations between utterances some characters can be split [4].

In the business process domain physical actions and messages convey information between participants. An initial proposal can be triggered by a certain action and later on be finished by another action. During a multi-party business collaboration, several proposals are initiated by different business parties. Each of them is followed and eventually finished by some action. Actions are thus sorted into different commitments.

For example, based on the collaboration structure model in Figure 3, an insurant phones a call center for a claim. Action A_phoneClaim triggered a conversation between the insurant and the call center to deal with the claim. Actions A_sendInfo and A_assignGarage follow, and action A_notifyClaim finishes the conversation between the insurant and the call center. Actions A_phoneClaim, A_sendInfo, A_assignGarage and A_notifyClaim are sorted within a commitment which records obligations of the call center.

Commitments. In this paper, a *commitment* is a guarantee by one party towards another party that some action sequences shall be executed completely provided that some "trigger", "involve", or "finish" action happens and all involved parties fulfill their side of the transaction [5], [6], [7]. To finish a commitment, more than one party must finish relevant actions.

We continue to use the case of which the collaboration structure model is presented in Figure 3. In Table 1 six commitments are identified according to the model of speech acts and repartee.

Table 1. Commitments, Actions and Action abbreviations

Commitment	Classification of Actions and Commitments			Labels
	Trigger	Involve	Finish	
C_phoneService (PS)	A_phoneClaim			PS.1
		A_sendInfo		PS.2
		A_assignGarage		PS.3
			A_notifyClaim	PS.4, CF.1, DS1
C_repairService (RS)	A_sendCar			RS.1
		A_estimateRepairCost		RS.2
	A_agreeRepairCar			RS.3, DS.7
			A_repairCar	RS.4, DS.8
C_claimForm (CF)	A_notifyClaim			CF.1, PS.4
		A_sendClaimForm		CF.2
			A_returnClaimForm	CF.3, PR.2
C_dailyService (DS)	A_notifyClaim			DS.1, PS.4, CF.1
		A_forwardClaim		DS.2
		A_contactGarage		DS.3
		A_sendRepairCost		DS.4
		A_assignAssessor		DS.5, IC.1
		A_sendNewRepairCost		DS.6, IC.3
			A_agreeRepairCar	DS.7, RS.3
	A_repairCar			DS.8, RS.4
		A_sendInvoices		DS.9
			A_forwardInvoices	DS.10, PR.1
C_inspectCar (IC)	A_assignAssessor			IC.1,DS.4
		A_inspectCar		IC.2
			A_sendNewRepairCost	IC.3,DS.5
C_payRepairCost (PR)	A_forwardInvoices			PR.1, DS.10
	A_returnClaimForm			PR.2, CF.3
			A_payRepairCost	PR.3

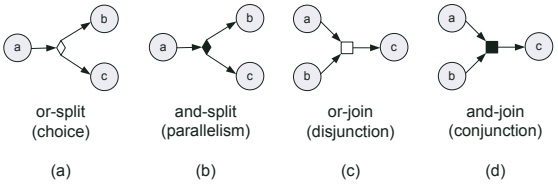


Fig. 4. Connector Representation

A multi-party business collaboration consists of a set of commitments. A collaborating party can thus be involved in different commitments playing different roles and an action may be involved in more than one commitment.

It is difficult to represent commitments graphically. In our commitment model, a *party* is represented as a rectangle with a name. A *node* denotes a role which should stay in a rectangle. A *commitment* is indicated by a set of nodes (or commitment connectors) and a set of arrows.

Commitments can be linked by causality, or-split, and-split, or-join, and and-join as Figure 4. The causality relation is represented by an arrow from one node to another node. The "or-split" relation is represented by an *empty-diamond*, which means that a commitment from a role triggers exactly one of multiple commitments from other roles (see Figure 4(a)). The "and-split" relation is expressed by a *solid-diamond*, which means that a commitment from a role triggers other multiple commitments (see Figure 4(b)). The "or-join" relation is denoted by an *empty-box*, which means that one of multiple commitment triggers another commitment (see Figure 4(c)). The "and-join" relation is shown by a *solid-box*, which means that multiple commitments together trigger a commitment (see Figure 4(d)).

Figure 5 depicts the five parties and six commitments of the case in Figure 3. For example, in party "insurance company", the solid-diamond connects commitments C_phoneService, C_claimForm and C_dailyService. It means

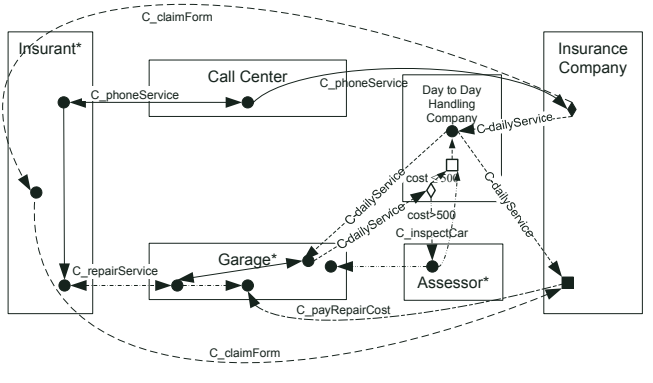


Fig. 5. Commitments of Multi-party Collaboration Model

that after commitment C_phoneService is fulfilled, both commitments C_claimForm and C_dailyService are triggered. A solid-box is also in party "insurance company", it connects commitments C_dailyService, C_claimForm and C_payRepairCost. It means that commitment C_payRepairCost will be performed after commitments C_dailyService and C_claimForm.

In the commitment model, we present the position of each commitment in terms of which parties and roles are involved, and which commitments are triggered, chose and paralleled by other commitments. In the commitment model, we provide an overview of each party's responsibilities. This is very important for both the business side and the IT side as it helps creating a common understanding. In the next section, the behavior of the parties is modeled.

3.3 Modeling Behaviors of the Parties

In multi-party business collaboration modeling, behavior is modeled as inter- or intra- organizational business processes. The vertical dimension is the time axis; time proceeds down the page. Each party is represented by a vertical column. An action is the atomic unit of behavior. The causal ordering between actions is modeled by conjunctive and disconjunctive "splits" and "joins".

Each party's behavior is determined by three parameters as found from the business collaboration. The inputs and outputs of a party are domain related. The rules of a party in our model are specified using predicate logic. The input parameter specifies the actions that this party expects to be involved in as object, while the output parameter specifies results of the action. When a party attempts to execute an action, it first checks whether the current input can trigger this action and subsequently generates the output which may be checked against the possible output.

In Figure 6, an action is noted as a table-box where the first column shows the action label which makes it possible to determine the commitment from this action label by looking it up in Table 1; the rest column shows the parameters involved in this action.

The diagram includes a multi-step interaction between participants. It clearly shows which parties will communicate with other parties for which matters. From the insurant's perspective, he only has contact with the call center, the assigned garage and the insurance company. From the insurance company's view, it receives and forwards the claim to the day to day handling company, sends the claim form to the policyholder, and finally pays the repair costs to the garage.

Each party parameter is also included in the diagram. According to the party properties, each party can determine which actions should or may occur. For example, after having received an input from the party property "Records2", the insurance company will perform actions A_sendClaimForm (as $CF2$) and A_forwardClaim (as $DS2$) according to rules "Records2 $\rightarrow$ CF_2" and "Records2 $\rightarrow$ DS_2 " respectively that are joined by an "and-join" parameter.

As time passes (from top to down) and satisfying the party parameters, each participant takes actions while the business process is moving forward. A complex multi-party business process is divided into multiple commitments.

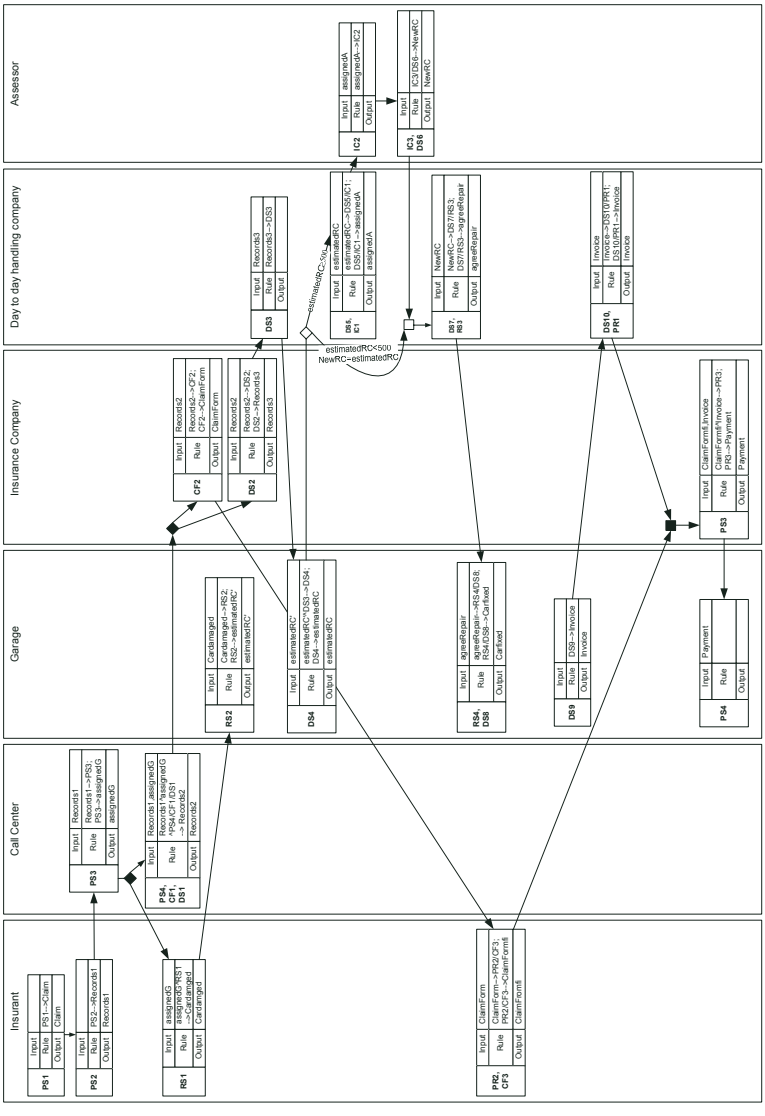


Fig. 6. Interactions between parties

Furthermore, business process steps are represented as the actions exchanged between the parties.

4 Related Work

There are few works concerning multi-party business collaborations. We provide the limitation of UML and Petri Net for multi-party business collaboration mod-

eling, as well as other models such as ebXML BPSS, web services choreography and SAP C-Business Scenarios.

According to [8], the UML is not suitable for modeling business. The UML does not support well all the concepts needed for business collaborations. Modeling business collaborations is mainly concerned with what happens at the business level and how it is organized. First, the UML does not support expressing responsibilities. In the business world, parties permit the commitments to each other to execute a business activity. In our approach, the commitment model represents collaborations between business parties and provides the relations of the commitments. Second, for business party behavior, business processes are normally not confined to single actions. Therefore, state diagrams (also for Petri Net) are not really useful here. Most business collaborators do not reason about processes in terms of states but rather in terms of the activities performed and results produced. Although UML can be extended with stereotypes and profiles, the stereotyping mechanism just makes it possible to introduce new subtypes of existing concepts and profiles cannot introduce new concepts either [9].

Using the Petri Nets to specify a multi-party business collaboration process, the amount of states of the Petri Net can be significantly increased. Especially, because the multi-party business collaboration process focuses on when and who did, is doing or will do what. A Petri Net representation can be too trivial, even by using state-based workflow patterns [10] because of a big amount of possible combinations of multi-party's behavior.

Two models for ebXML BPSS multi-party collaboration and web services choreography are presented in [11], [12] respectively. Other research [13], [14] on multi-party collaboration tries to break down a multi-party collaboration into a number of bilateral relations. A principle cause behind this is that current e-commerce environments only support bilateral executions. In some simple cases, the approach to support multi-party collaboration execution in current e-commerce environments is to assume the whole business process runs correctly according to a number of bilateral relations. However, in complicated multi-party collaborations this conversion results in information of relations being lost or hidden. Consequently this option to split the multi-party collaborations up into several two-party relations will not work for these complex multi-party collaborations.

SAP's collaborative business scenarios describe inter-enterprise business processes from three different perspectives [15], namely business view, interaction view and component view. The purpose of the business view is showing the business advantages of implementing a collaborative business scenario. Business relations per se are out of the scope of our research through. The interaction view describes the process design and detailed dependency relationship between the different activities and responsibilities of the participants. It is too simple to describe the relationship like the action relations with conjunctive and disconjunctive "splits" and "joins". The component view describes the logical application components needed to support the business process. Different channels in a collaboration structure model can determine different ways to implement

a multi-party collaboration. Commitment model and interaction model provide enough details of interactions between multi-parties. Those three collaboration models can easily map into a component level model by using specifically software implementation packages.

5 Conclusions

We have present multi-party business collaboration models from three perspectives. At the collaboration structure model, we provide a view of how business parties are linked. Different links can determine different ways of collaboration. In the commitment model, the responsibilities of all involved parties are presented. Finally, the behavior model provides details of commitment fulfillment. Further research has to map our multi-party business collaboration model to specific implementations like SAP or BAAR' ERP systems. This would allow the semantics of the web of collaborating parties to be validated.

References

1. Austin, J.: How to do things with words. 2 edn. Oxford University Press (1975)
2. Longacre, R.: An anatomy of speech notions. Lisse: de Ridder (1976)
3. Van Dyke Parunak, H.: Visualizing agent conversations: Using enhanced dooley graphs for agent design and analysis. In: The Second International Conference on Multi-Agent System (ICMAS'96). (1996)
4. Dooley, R.: Repartee as a Graph. Appendix B in [2] (1976)
5. Xu, L., Jeusfeld, M.A.: Pro-active monitoring of electronic contracts. In: The 15th Conference On Advanced Information Systems Engineering in Lecture Notes of Computer Science. Volume 2681., Springer-Verlag (2003) 584–600
6. Xu, L.: Monitoring Multi-party Contracts for E-business. PhD thesis, Tilburg University (2004)
7. Xu, L., Jeusfeld, M.A.: Detecting violators of multi-party contracts. In: The Conference on CoopIS/DOA/ODBASE in Lecture Notes of Computer Science. Volume 3290., Springer-Verlag (2004) 526–543
8. Steen, M., Lankhorst, M., van de Wetering, R.: Modelling networked enterprises. In: the sixth international Enterprise distributed Object Computing Conference (EDOC'02). (2002)
9. Henderson-Sellers, B.: Some problems with the uml v1.3 metamodel. In: the 34th Annual Hawaii International Conference on Systems Sciences (HICSS-34). (2001)
10. van der Aalst, W., ter Hofstede, A., Kiepuszewski, B., Barros., A.: Workflow patterns. Distributed and Parallel Databases (2003) 5-51
11. Dubray, J.J.: A new model for ebxml bpss multi-party collaborations and web services choreography (2002)
12. Webber, D.: The benefits of ebxml for e-business. In: International Conference on XML (XML'04). (2004)
13. Dubray, J.J.: A new model for ebxml bpss multi-party collaborations and web services choreography (2002) `http://www.ebpml.org/ebpml.doc`.
14. Haugen, B.: Multi-party electronic business transactions (2002) `http://www.supplychainlinks.com/MultiPartyBusinessTransactions.PDF`.
15. SAP: mysap.com collaborative business scenarios,whitepaper. In: Walldorf. (2000)

A Framework for Task Retrieval in Task-Oriented Service Navigation System

Yusuke Fukazawa, Takefumi Naganuma, Kunihiro Fujii, and Shoji Kurakake

Network Laboratories, NTT DoCoMo, Inc.,
NTT DoCoMo R&D Center, 3-5 Hikarino-oka, Yokosuka, Kanagawa, 239-8536, Japan
{y-fukazawa, naganuma, kunihi_f, kurakake}@netlab.nttdocomo.co.jp

Abstract. Mobile access to the Internet is increasing drastically, and this is raising the importance of information retrieval via mobile units. We have developed a task-oriented service navigation system[6] that allows a user to find the mobile contents desired from the viewpoint of the task that the user wants to do. However, the user is still faced with the problem of having to select the most appropriate task from among the vast number of task candidates; this is difficult due to the fact that mobile devices have several limitations such as small displays and poor input methods. This paper tackles this issue by proposing a framework for retrieving only those tasks that suit the abstraction level of the user's intention. If the user has settled on a specific object, the abstraction level is concrete, and tasks related to the handling of the specific object are selected; if not, tasks related to general objects are selected. Finally, we introduce two task retrieval applications that realize the proposed framework. By using this framework, we can reduce the number of retrieved tasks irrelevant to the user; simulations show that roughly 30% fewer tasks are displayed to the user as retrieval results.

1 Introduction

The mobile Internet web is expanding drastically from various viewpoints, such as the number of subscribers and the volume of mobile contents[1][2]. i-mode, which is NTT DoCoMo's mobile Internet access service and also a trademark and service mark owned by NTT DoCoMo, is one of the most successful mobile service in the world[3], with over 44 million subscribers and counting[4]. i-mode users can access more than 4,800 approved (by NTT DoCoMo) sites dedicated for i-mode users and over 85,000 independent sites specifically designed for the use by i-mode handset[5]. As the mobile Internet gains in popularity, information retrieval through the mobile web must be made easier and more efficient.

We have developed a task-oriented service navigation system[6] that allows a user to find the mobile contents desired from the viewpoint of what user wants to do. For this system, we modeled the mobile user's daily activities as user tasks and associated those tasks with relevant mobile services that assist the user's activities. Tasks are categorized according to the concept of places, called domains in the task-model, where each task is likely to be performed. This is

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 876–885, 2005.
© Springer-Verlag Berlin Heidelberg 2005

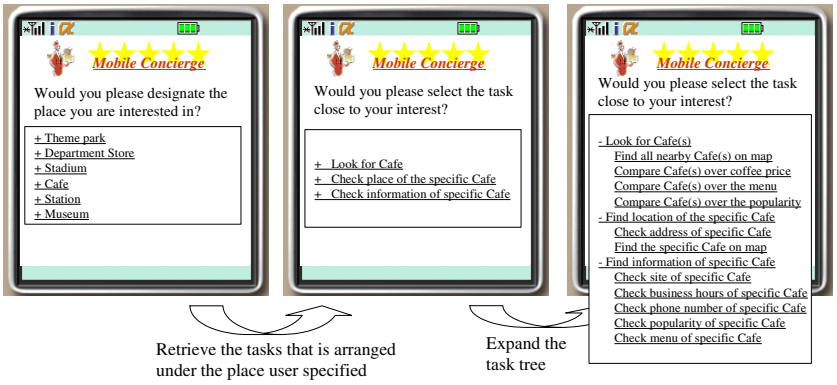


Fig. 1. Simulation view of task-oriented service navigation system

efficient because a mobile service is used most often outside the home. At first, a user selects the place user is interested in. Next, the task list for the selected place is retrieved and sent to the mobile device. An example is given in Fig.1. If the user wants to drink coffee, when in an unfamiliar location, the user selects Cafe domain, and the tasks associated with Cafe domain are shown. The user is shown tasks related to cafes and selects the most appropriate task. The task-model consists of multiple layers, and the user can uncover deeper layers by clicking a task node on the display. After selecting a task, the user is provided with a list of contents whose functions are relevant to the selected task. The task-oriented approach, which arranges the mobile contents according to the function of the contents instead of the name of service category, allows the user to find the service desired by selecting the task that uaer wants to do.

A remaining problem is that the user must select the most appropriate task from the vast number of task candidates displayed. Unfortunately, current devices have several limitations such as small displays and poor I/O operations. This paper tackles this issue by proposing a framework for retrieving tasks and forming a task list that suits the user's intention. The problem of forming the items of the menu list to suit the user's intention has been tackled in the field of personalizing mobile sites.

1.1 Past Approaches to Personalizing Mobile Sites

Most prior research takes the approach of utilizing the user's profile or usage history to arrange the contents of a web site to suit the user's intention. Corin et al. proposed the "Web site personalizer", which can make frequently visited destinations easier to find by making the anchor text bold face[7]. It also highlights a link that interests the visitor, and elides uninteresting links based on the usage history. Christoforos et al. proposed "mPERSONA", a personalized mobile portal[8]. mPERSONA transforms the content provider's Metadata Tree, which is the semantic structure of the content of the provider's site, into the user's

personalized portal based on usage history by discarding both unneeded links and contents.

The published approaches to personalization are not always successful in extracting and reflecting the user's current intention. They presume the user's intention by comparing current and past contexts, and this approach is not powerful enough to select items relevant to the user's current intention. Because a web site typically contains different kind of items and the selection of different kind of items may require considering different aspects of the user's intention, it is very unlikely that there exists one universal method that can be applied to all items in a large web site.

In the case of the task-oriented service navigation system, we do not directly select services, but do it through task concept lists that are carefully designed based on the same principle. This is why we believe it possible to come up with one universal method that can make service selection reflect the user's intention. In this paper, we focus on the abstraction level of the user's intention. It is a measure of the extent to which the user has settled on the target object of the task.

The abstraction level of the user's intention is easily obtained by asking simple questions like "Do you have a specific Cafe in mind?" or analyzing the query input by the user. If the user's query contains words that only mention the class of an object, the abstraction level is high, and if the query contains specific objects, the abstraction level is low.

In addition, the task can be categorized into two types, one type corresponds to abstract intention and the other corresponds to specific intention. Therefore, using this abstraction level should make task retrieval far more efficient by eliminating those tasks that do not match the abstraction level of the user's current intention.

In the following section, we propose a framework for task retrieval that is based on the user's abstraction level. In Section 3, we show two applications that utilize the proposed framework. In Section 4, we conclude this paper. We abbreviate "abstraction level of user's intention to "abstraction level" hereafter.

2 Framework for Task Retrieval According to Abstraction Level

2.1 Proposed Framework for Task Retrieval

In this section, we propose a framework to retrieve the tasks that match the abstraction level. The proposed framework is shown in the Fig.2. This framework consists of three parts: mobile handset, task knowledge base, and task retrieval server. A mobile user selects the place and then selects the task closest to the user's intention from the task list on the mobile handset. The task retrieval server acquires the abstraction level by asking the user some questions, retrieves the task lists that match the abstraction level from the task knowledge base, and sends the task lists to the mobile handset. The task knowledge base stores the task model.

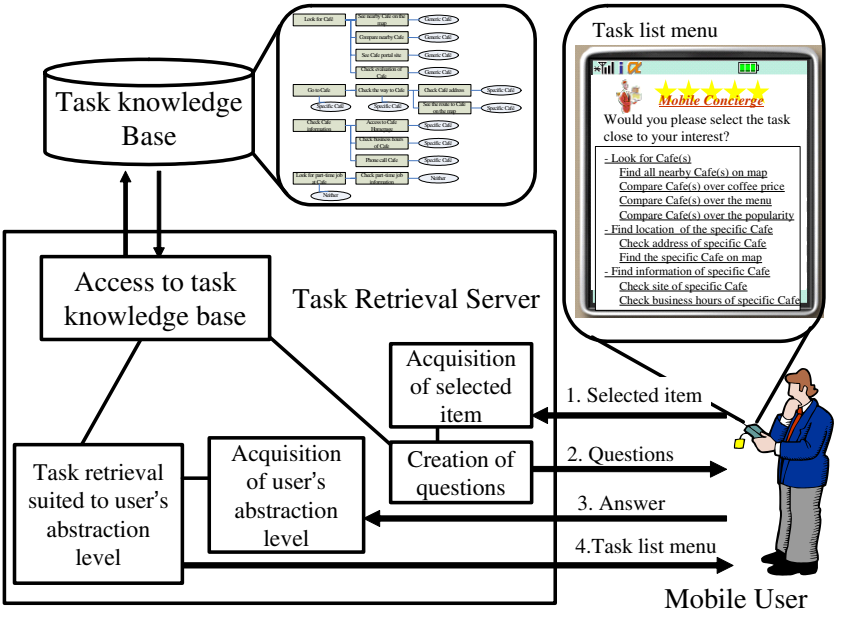


Fig. 2. A Framework for task retrieval based on abstraction level of user's intention

The general flow of this framework is as follows. First, the mobile user selects the task that is closest to the user's intention from the task list, and the ID of the selected task is sent to the task retrieval server. The set of sub-tasks of the selected task is retrieved from the task knowledge base and passed to the task retrieval server. One or more questions that aim to acquire the abstraction level are created from the annotation data of the sub-tasks. The questions are put to the mobile user and the abstraction level is acquired from the user's answers. Sub-tasks that match the abstraction level are extracted from the retrieved set of sub-tasks. The extracted sub-tasks are sent to the mobile handset, and then the task list is created giving the retrieved sub-tasks high priority.

In the case of task retrieval just after selecting the item of the top menu, which is the list of places, the sub-tasks will be the tasks that are associated with the selected place. We apply the same flow as described above for this case.

2.2 Association of the Abstraction Level of User's Intention with Corresponding Type of Task

This section explains how to associate the abstraction level of user's intention with the corresponding abstraction level of the task. We first define the type of the target object of the task according to its abstraction level as follows:

1. Task whose target object is generic object such as "store"
2. Task whose target object is specific object such as "XYZ Cafe"

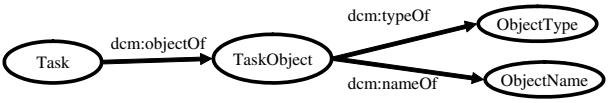


Fig. 3. Description model of annotating the abstraction level of the object of the task, "ObjectType", and the name of the object, "ObjectName"

```
<owl:ObjectProperty rdf:ID="objectOf"/>
  <rdfs:domain rdf:resource="#Task"/>
  <rdfs:range rdf:resource="#TaskObject"/>
</owl:DatatypeProperty>
<owl:Class rdf:ID="TaskObject"/>
<owl:ObjectProperty rdf:ID="typeOf">
  <rdfs:domain rdf:resource="#TaskObject"/>
  <rdfs:range rdf:resource="#objectType"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="nameOf">
  <rdfs:domain rdf:resource="#TaskObject"/>
</owl:ObjectProperty>
<owl:Class rdf:ID="objectType">
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:ID="SpecificObject "/>
    <owl:Class rdf:ID="GenericObject"/>
  </owl:unionOf>
</owl:Class>
```

```
<?xml version="1.0" encoding="Shift_JIS"?>
<!DOCTYPE uridef[
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns">
<!ENTITY service "http://www.daml.org/service/owl-s/1.0/Service.owl">
<!ENTITY profile "http://www.daml.org/service/owl-s/1.0/Profile.owl">
<!ENTITY dcm "http://nttdocomo.com/task.owl">]>
<rdf:RDF xmlns:rdf = "&rdf," xmlns:service = "&service;"
xmlns:profile = "&profile;" xmlns:dcm = "&dcm;">
<service:Service rdf:ID="Task00000019">
<service:presents><profile:Profile>
<profile:serviceName>Look for Cafe</profile:serviceName>
<dcm:objectOf rdf:type="Resource">
<dcm:typeOf rdf:resource="&dcm;#GenericObject"/>
<dcm:nameOf>Cafe</dcm:nameOf>
</dcm:objectOf>
</profile:Profile></service:presents></service:Service>
</rdf:RDF>
```

Fig. 4. Left: OWL schema based on the description model shown in Fig.3, right: Description of the task "Look for Cafe" using OWL schema of Fig.4

We annotate the task description according to the type of the target object of the task by the tag "ObjectType". The tasks whose object is Type (1) are given the annotation "GenericObject" as "ObjectType", while the tasks whose object is Type (2) are given the annotation "SpecificObject". In addition, the task description is given the annotation "ObjectName", that is the name of the object. Note that the task-model simplifies the Type (2) annotation descriptions in that it uses the category name of the specific object as the "ObjectName" instead of the actual name of the specific object. For example, given the user's intention "Find the location of the "XYZ Cafe"", the system identifies the "ObjectName" as "Cafe", which is the category name of "XYZ Cafe".

A description model and corresponding OWL schema are shown in Fig.3 and the left side of Fig.4 respectively. The right side of Fig.4 shows the description of the task "Look for Cafe" using the OWL schema on the left side of Fig.4.

Next, we associate the abstraction level of the user's intention with the corresponding type of the task. There are two abstraction levels for user's intention: concrete and abstract. Concrete level indicates that the user has decided the specific object as the target of the task, and abstract level indicates that the user has not yet decided the specific object.

If the user is interested in a specific Cafe such as "ABC Cafe" when user selects the Cafe for place, tasks whose "ObjectType"="SpecificObject" such as "Find location of the <specific> Cafe" are retrieved. This is because user has already decided the specific target of the task, and user is considered not to be

interested in the tasks whose "ObjectType"="GenericObject" such as "Look for Cafe(s)". On the other hand, if the user has yet to decide the specific target of the task, tasks whose "ObjectType"="GenericObject" such as "Look for Cafe(s)" are retrieved.

2.3 Task Retrieval According to the Abstraction Level of User's Intention

This section describes a method that acquires the abstraction level of the user's intention, and a method for task retrieval based on the acquired abstraction level. To acquire the abstraction level, the system uses questions like "Have you settled on the specific "ObjectName"? " for each "ObjectName" of the sub-task of selected task or item. The system issues such questions when the user selects a task or place listed on the menu. The abstraction level is acquired from the user's answer.

As described in the previous section, if the user's abstraction level is judged as concrete, tasks whose "ObjectType"="SpecificObject" are retrieved. If the abstraction level is abstract, tasks whose "ObjectType"="GenericObject" are retrieved. The retrieved tasks are sent to the mobile handset, and the set of tasks that match the abstraction level are shown together under the parent task node on the display of the mobile handset.

3 Applications Realizing Proposed Framework

In this section, we introduce two applications: user-query-based task retrieval and map-based task retrieval. Both utilize the framework described in the previous section. We consider tasks associated with the place "Cafe" and the "ObjectName" annotated to the tasks is "Cafe". The results of categorizing the task according to the framework described in the previous section are shown in Fig.5.

3.1 Simulation of User-Query-Based Task Retrieval

The user-query based approach can abbreviate the process of both selecting the place and answering questions raised by the proposed framework. It allows us to acquire the place and the abstraction level from the user's query. First, the user is asked to input a text query that is related to the place such as "bus station", "Yankee's stadium" or "Chinese theater" etc., in order to distinguish the abstraction level. Next, the query is categorized as to whether it contains a proper noun or a generic noun by referring to a thesaurus. If the user query exists in the thesaurus, the query is judged as a general noun. On the other hand, there is no such term in the thesaurus; the query is judged as a proper noun.

If the query is a general noun, the thesaurus is referred to again in order to convert the user query into a place name that is used in the task-model. For example, the query "Coffee shop" is converted into "Cafe" since the latter is used in the task-model. After that, tasks whose "ObjectType"="GenericObject" are retrieved. Finally, retrieved tasks are sent to the mobile device and arranged

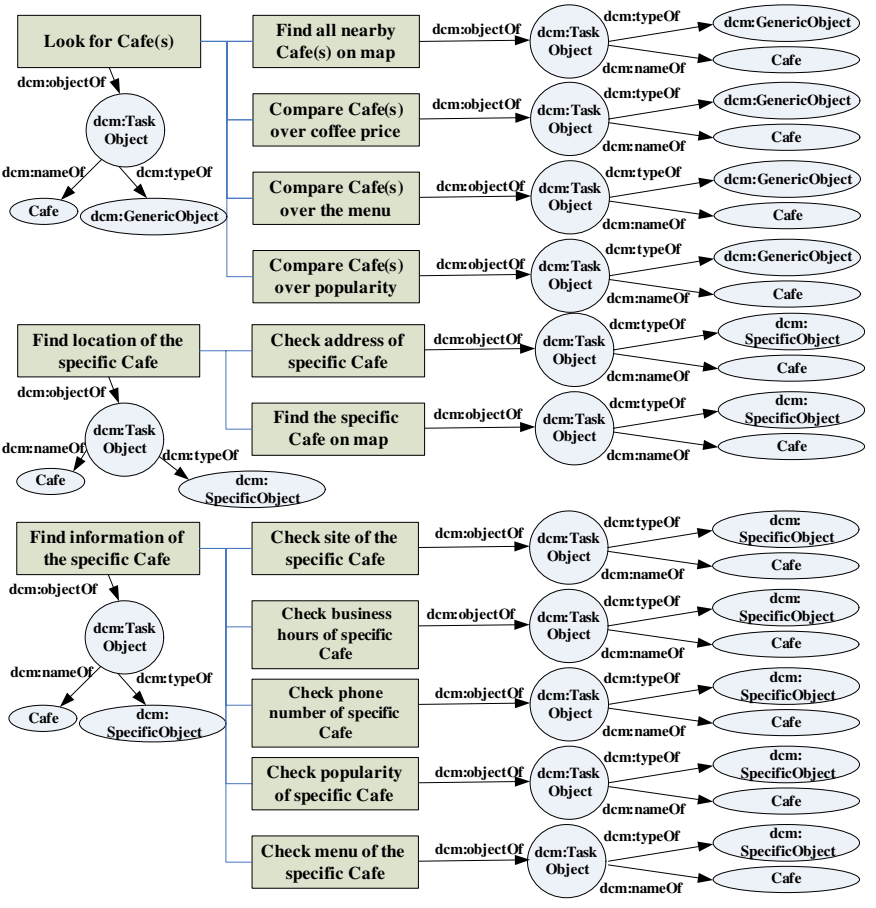


Fig. 5. Description model of the tasks associated with place "Cafe"

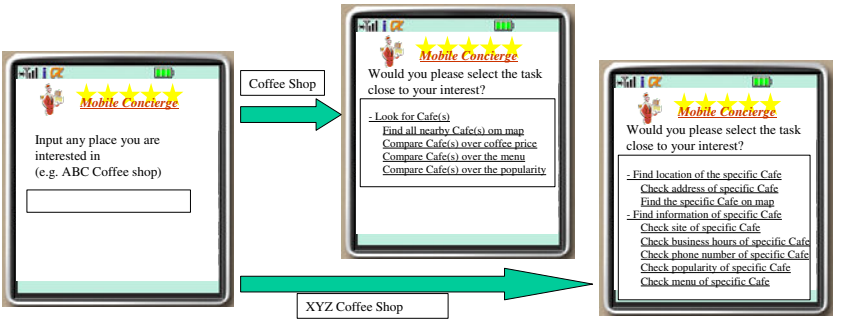


Fig. 6. Simulation of user-query-based task retrieval

Table 1. General-proper noun conversion table

Shop name (proper noun)	Category name (generic noun used in task-model)
Coffee shop A	Cafe
Tokyo Station	station
Narita Air port	Airport

with children node expansion. This flow is shown in Fig.6. From this figure, the number of tasks retrieved is reduced to 5 tasks compared to the 14 tasks shown in Fig.1:right.

If the user's query contains a proper noun, the general-proper noun conversion table shown in Table.1, is accessed to in order to convert the query into a place name used in the task model. For example, "XYZ Coffee Shop" is converted to "Cafe". After that, the tasks whose "ObjectType"="SpecificObject" are retrieved. Finally, retrieved tasks are sent to the mobile device and arranged with children node expansion. This flow is shown in Fig.6. From this figure, the number of tasks retrieved is reduced to 9 tasks compared to the 14 tasks shown in Fig.1:right.

By using this framework, we can reduce the burden placed on the user; the number of tasks retrieved is reduced by at least 30%. As can be seen from the above discussion, the user-query-based task retrieval model can retrieve the tasks that suit the abstraction level of the user's intention.

3.2 Simulation of Map-Based Task Retrieval

This section proposes a map-based application for task retrieval. The map-based approach can improve the efficiency of selecting a place by showing a map of the user's current location instead of a list of category names.

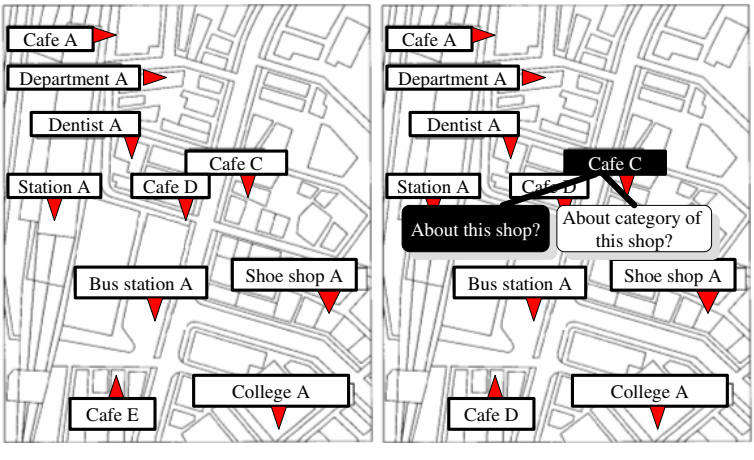


Fig. 7. Simulation of map-based task retrieval (left: the map with shop and station name indicated by black boxes, right: questioning the user as whether the user wants to search for the clicked shop or the category of the shop)

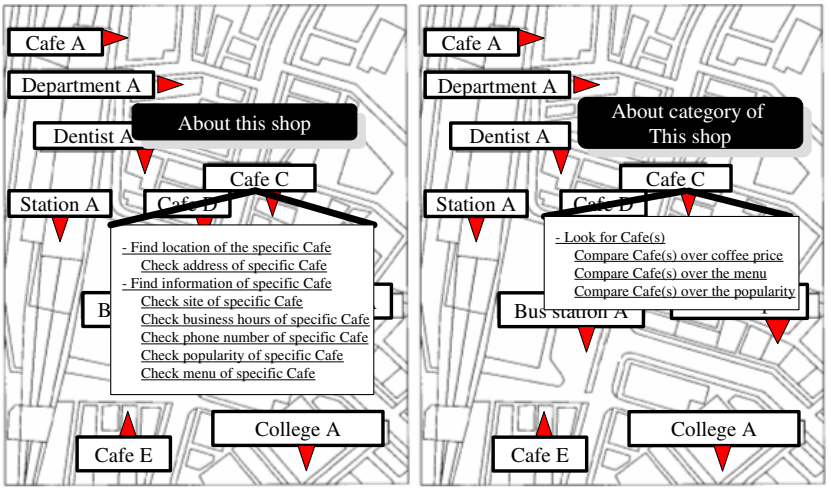


Fig. 8. Results of task retrieval for both abstraction levels in the simulation of map-based task retrieval (left: concrete level, right: abstract level)

At first, the local map is shown on the display using a location information service e.g. GPS. The simulation provided a map that offered three kinds of information: shop name, general category name of the shop, and location information (longitude and latitude). For example, the shop "ABC Coffee Shop" has the following entries; shop name: "ABC Coffee Shop", general category name: "Cafe", and GPS data (34,118). One district in Japan is shown on the left side of Fig.7. The shops and other places on the map are indicated by the black boxes.

Next, the user clicks the black box of a shop or other place that the user is interested in. At this point, it is not clear whether the user is interested in the actual shop (concrete level), or its category (abstract level).

Therefore, user is asked whether user is interested in the clicked shop or the category of the clicked shop in order to distinguish the abstraction level as shown in Fig.7: right side. If the user selects "about this shop", the abstraction level is concrete, and tasks whose "ObjectType"="SpecificObject" are retrieved. If the user selects "about the category of this shop", the abstraction level is abstract, and tasks whose "ObjectType"="GenericObject" are retrieved. Finally, retrieved tasks are sent to the mobile device and arranged in the same manner as in the application of user-query-based task retrieval. Fig.8 shows the results (left: concrete level, right: abstract level). As can be seen, the map-based task retrieval model can also retrieve tasks that suit the abstraction level of the user's intention.

4 Conclusion

In this paper, we proposed a framework to retrieve tasks that suit the abstraction level of the user's intention for task-oriented mobile content retrieval. This

framework includes a method of using questions to acquire the abstraction level of user's intention, and a method to retrieve tasks based on the abstraction level. Along with this framework, we categorized the tasks according to the abstraction level of the target object of the task, which we describe using an OWL schema. In addition, we introduced two applications: user-query-based task retrieval and map-based task retrieval. Both are based on the proposed framework. By using this framework, we could reduce the tasks irrelevant to the user by considering the user's abstraction level; in simulations, the number of retrieved tasks presented to the user was reduced by at least 30%.

In future work, we plan to extend this framework so as to utilize the user's context or person-specific information, not considered in this paper, for task retrieval.

References

1. A. Serenko and N. Bontis: A model of user adoption of mobile portals. Quarterly Journal of Electronic Commerce 4(1): 69-98, 2004.
2. Daniel Olsson: Mobile Portal Content. the 10th European Conference on Information Systems, 2002.
3. T. Hart and P. Leet: Research Brief - Worldwide Wireless Portal Comparisons, GartnerGroup, 2000.
4. http://www.nttdocomo.co.jp/english/corporate/investor_relations/business/imode use_e.html
5. http://www.nttdocomo.co.jp/english/corporate/investor_relations/business/contr_ e.html
6. T. Naganuma and S. Kurakake: A Task Oriented Approach To Service Retrieval in Mobile Computing Environment. In Proceedings of Artificial Intelligence and Applications, 2004.
7. Corin R. Anderson, Pedro Domingos, and Daniel S. Weld: Personalizing Web Sites for Mobile Users. In Proceedings of the 10th International Conference on World Wide Web, 565-575, 2001.
8. C. Panayiotou, G. Samaras: mPERSONA: personalized portals for the wireless user: An agent approach, Mobile Networks and Applications, 663-677, Vol. 9 , Issue 6, 2004.

Realizing Added Value with Semantic Web

Dariusz Kłeczek, Tomasz Jaśkowski, Rafał Małanij, Edyta Rowińska, and Marcin Wasilewski

Scientific Association for Computer Science, Warsaw School of Economics
{dk23364, tj3444, rm28708, er28909, mwasil}@sgh.waw.pl

Abstract. The Business Case for Semantic Web requires reports on its usefulness in real-life scenarios. In this study we introduce a framework for analysing potential application scenarios of the Semantic Web, which is based on the concept of added value. Based on this analysis, we have formulated a hypothetical life cycle of Semantic Web technologies in corporate IT infrastructures. It identifies corporate knowledge management as the application area, where a killer application is most likely to be developed. As a proof of concept the design, deployment and evaluation of a Skills Management system is presented.

1 Introduction

As the Semantic Web (SW) [4] technologies mature and the focus of their application shifts from academic to industrial field, business potential of SW has increasingly been subject of research. In this paper, we introduce a framework for classifying the application areas of SW, which is based on the concept of added value. This framework has enabled us to focus on Knowledge Management as the area, where SW technologies are most likely to settle initially. Once established in the enterprise IT architecture, they are more likely to disperse into other areas.

As a proof of concept we have designed, deployed and evaluated a Skills Management system – TeamBuS (which is an abbreviation for Team Building System) in a scientific organization – SKNI – associating students interested in Computer Science at the Warsaw School of Economics. SKNI is part of a virtual community of ca. 70 organizations gathering over 1000 students and realizing almost 200 scientific projects per year. We have identified this as a perfect testing ground for the evaluation of a SW application.

TeamBuS provides three basic functions: staffing projects with persons who best match the requirements, finding experts in a given field and analysing the knowledge available (or the knowledge gap) in an organization. Its data is based on RDF and ontologies, and the end user interface is realised with the SW portal Ontoviews [25].

In the remainder of this paper we first give an overview of related work. The next chapter introduces the added value framework, which is followed by examples of SW application scenarios it can help to analyse. Chapter 5 presents models of adoption of SW in corporations, while chapter 6 describes a case study we have conducted. Conclusion and an overview of future work follow in chapter 7.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 886 – 895, 2005.
© Springer-Verlag Berlin Heidelberg 2005

2 Previous Research

It is widely acknowledged, that for the SW vision [4] to realize, a critical mass of semantically annotated data has to be provided [16]. Haustein [16] perceives redundancy and the costs it implies as the biggest obstacle in adopting SW technologies. In order to lower the barrier for entry, the support of tools building on existing knowledge, avoiding redundancy, providing additional value and gain needed to outweigh extra cost are proposed. There exist also doubts in the realization of the SW vision, instead the application of SW technologies on a smaller scale is suggested [19]. Nevertheless, a large number of prototype and commercial systems have been developed, most of them identifying the benefits associated with the application of SW technologies [5,8,9,28,32].

Classifications of SW application scenarios are also present in the literature [11,12,14,15], usually identifying Knowledge Management and Electronic Commerce as two big application areas. Following this distinction, multiple researchers have analysed potential benefits associated with the application of SW in these areas [30,31,36]. In this study we provide a different framework, allowing for a more fine-grained classification of SW applications.

It is common to see Knowledge Management as the place for a “killer application”, which migh leverage the usage and acceptance of SW technologies [2,11,14]. In this area, Skills Management has also been recognized and provided with devoted, ontology-based systems [1,22]. The idea of employing a semantic portal for the dissemination of semantic data is also well grounded in the SW community [23,24,25]. Furthermore, there exist concepts and examples for the evaluation of SW technologies [10,18,33,34]. The environment of distributed virtual organizations as the target of SW application has been examined for example in [18].

In this context, our approach is innovative in the way, that it provides a method for prediction of the dispersion of semantic data and applications inside and outside of enterprises. The realization and evaluation of TeamBuS has proven the concept and has enriched us with additional “lessons learned”. Firstly, the successfull realization of a semantic application with open source tools can be of interest for the SME-sector. Secondly, the enabling position of SW in an environment of distributed virtual organizations has been identified.

3 Framework

Our analysis is based on the insight, that adoption of SW in enterprises should be regarded as a strategic decision. As such, it can be studied with appropriate tools, for example portfolio analysis.

Characteristic for SW is the dictinction between provider of semantically annotated data and beneficiary, who makes business use of the data. This feature moved us to distinguish between business applications which produce semantic data and applications which merely make use of external data. We assume that in the first case, SW is implemented inside a company, and in the latter case outside of it. Another distinction can be observed in the motivation of organizations adopting SW. It is natural that such decision has to bring added value, however its creation can take place either

inside an organization (value for the company) or outside of it (value for the customer and/or business partner).

These two dimensions – place of implementation and place, where added value is created – result in a matrix and we classify most popular SW application scenarios accordingly (see Fig. 1).

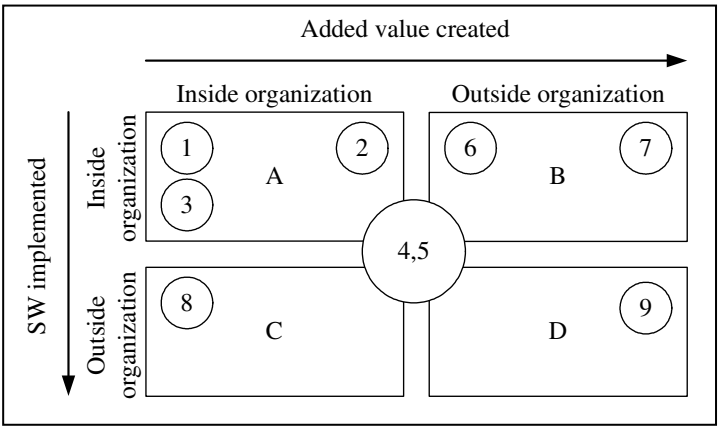


Fig. 1. Classification of SW application scenarios

4 Application Scenarios

We have distinguished nine SW application scenarios, two of which can be classified on the intersection of all four matrix fields (see Fig. 1).

EAI (1 in Fig. 1) covers the integration of enterprise application systems inside of a company, differently from B2B, which involves the integration of application systems from distinct companies [26]. The promise of SW lies in improving the efficiency through providing of ontologies to semantically describe data structures. Such annotation of data structures with metadata can be used by EAI technologies, resulting in the elimination of problems concerning the interpretation of internal structures of applications, which previously had to be solved by qualified analysts [7].

Knowledge Management (2 in Fig. 1) deals with acquiring, maintaining, and accessing knowledge of an organization [14]. According to [30], potential benefits of SW can be realized in three areas of Knowledge Management. Firstly, searching for knowledge with SW can lead to better precision and recall. Relationships can be analysed and complex queries with inference mechanisms can be formulated, which might lead to the discovery of not explicit relationships. Ontologies can be used for graphical navigation, and the navigation structures can be automatically generated. Secondly, presenting knowledge with SW can provide benefits through visualizing the results, presenting of related documents and semantic based Push-services. Thirdly, in the area of e-learning the SW can enable individual configuration of e-learning products and requirement-driven access to contents [30]. The realisation of potential benefits of the SW in Knowledge Management is, however, questionable.

Firstly, the implementation of SW technologies can be costly. It is not only the annotation of documents that has to be paid, but there are also the processes of construction, storing, aligning and maintaining of ontologies, which take up resources. Extremely important is in this context security. Lack of standards in this area can be seen as a serious obstacle stopping the deployment of SW technologies in corporate Knowledge Management.

Potential benefits of the SW can also be realized in the area of Computer Supported Cooperative Work (CSCW – 3 in Fig. 1). CSCW is understood as the use of Information Technology to support teamwork. There, an automation of routine tasks [17], an improvement in communication through the use of ontologies [20] and an augmentation of workflows through dynamically connecting software components, which are semantically annotated [27] can be achieved.

It is possible to use the SW to support negotiation processes (4 in Fig. 1), which are much better suited than fix price system to provide optimal allocation of resources. The transaction costs of negotiations can be diminished through the use of SW technologies and software agents.

B2B (5 in Fig. 1) is similar to EAI except for the fact, that application systems belonging to distinct companies are integrated. As with EAI, the application of the SW or Semantic Web Services can be beneficial. SW technologies can be also applied to automate the processes of discovery, description and access to applications of business partners, which are provided within the Semantic Web Services framework. A semantic, process oriented description of services (e.g. with RDF or OWL) allows their dynamic assembling and configuration by software agents to more complex Web services. For example, an agent can decide to use a more expensive, but faster service, whenever a project is delayed [35].

The consumers should profit from innovations, while being relieved of routine tasks by intelligent software agents. SW technology as an enabler (6 in Fig. 1) promises many new applications, especially in the areas of integration of electronic devices and of personal information services. Personal software agents should support their users by processing information, while performing search for information, negotiations and sometimes even decision making on their own. They should be able as well to manage new generations of household devices, such as refrigerators or microwave ovens.

Another potential benefit for the customers is personalisation (7 in Fig. 1). It means to provide users with individualised pages, based on some form of model of their preferences. The goal of this, for example in the application area of online-shopping-systems would be to create an impression of a local ‘corner shop’ [13].

Further potential benefits can be achieved through the combination of SW and Web Mining (8 in Fig. 1). The goal of Web Mining is “to develop methods and systems for discovering models of objects and processes on the World Wide Web and for web-based systems that show adaptive performance” [3]. Making use of semantics could greatly improve the results of Web Mining. Its potential application fields are, for example, Knowledge Management, e-commerce and e-learning.

The development and widening of SW technologies, which is predicted by many authors, can lead to a situation, where enterprises that do not offer semantically enriched products will not be able to face competition. The implementation of SW technologies through enterprises can be based on strategic reasons (9 in Fig. 1), in order to build up the necessary know-how.

5 Life Cycles

Based upon our framework, we have formulated a hypothesis regarding theoretical life cycle of SW technologies in an organization (see Fig. 2). The application of SW technologies should start in square D and then move to A, followed by B and C. In the first stage (square D), an organization builds up the know-how, for example by funding research in the area of the emerging technology, without producing any internal added value. In the second stage (square A), the technologies are implemented internally, producing added value inside the organization. This includes the implementation of either EAI, Knowledge Management, or CSCW. It seems, that at present Knowledge Management is the most mature application field of SW technologies. After the technology has settled inside organizations, this can be used to create cooperations, where multiple corporations apply the technologies to create added value. This is the case of both B2B and negotiations, which form the third stage of the model. While these stages provide the critical mass of semantically annotated data, the more "visionary" applications of SW can be realized in the fourth stage. It involves the scenarios classified in squares B and C: SW as enabler, personalisation, and Web Mining.

While this life cycle may be relevant rather for big companies, when only for the fact that it includes multiple SW application scenarios, several other life cycles can also be predicted, each focusing on different square of the grid. They can be depicted as investor, initiator, parasite and subordinate life cycles.

The focus of the *investor life cycle* lies in improving corporate efficiency through SW applications. Therefore, it starts in square A. In effect, other application scenarios may also be undertaken – an evolution to adopt SW in the areas of B2B, negotiations and serendipity (SW as enabler) is likely to happen.

The *initiator life cycle* involves the concentration on customer as the beneficiary of SW application. It includes the scenarios SW as enabler and personalisation classified in square B. It is connected with high amount of risk due to the fact, that a widespread adoption is usually necessary in order to provide the critical mass of data required for SW to pay off. A following adoption of other application scenarios is possible but not inevitable.

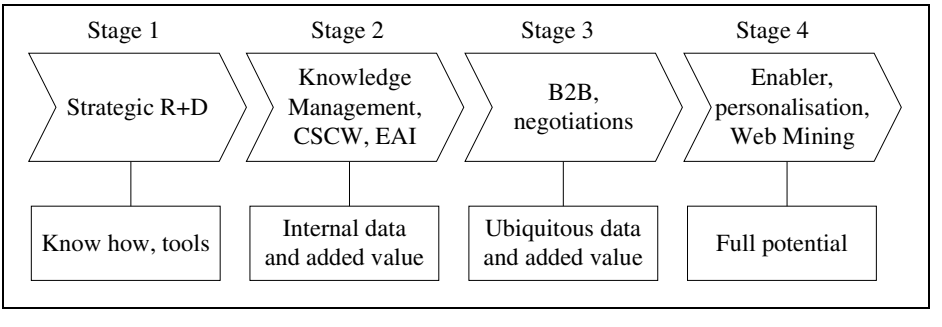


Fig. 2. Four stage model of the adoption of SW in corporations

The *parasite life cycle* takes advantage of already available semantic data. The existence of such data may move organizations to utilize it, for example with Web Mining (square C). In consequence, the generated know how may be used to implement other SW applications, for example these classified in squares A or B.

A company can be prompted to implement SW by business partners, who might use their position to initiate B2B or negotiation projects. This is the beginning of the *subordinate life cycle.* Such companies also generate know how applicable in other areas, which creates the possibility of adopting every other SW application.

In the following case study we have concentrated on our first hypothesis. In order to evaluate it, we have designed, implemented and deployed TeamBuS, which can be classified in the area of Knowledge Management. The evaluation of the system included the assessment of probability, that the transition to following phases of the life cycle would take place.

6 Case Study

As a proof of concept for our hypothesis we have decided to follow the four stages life cycle of SW adoption in organizations. We have discovered the cluster of ca. 70 students' organizations at the Warsaw School of Economics as a very promising field for our research. These organizations, each comprising of 10 to over 100 members, cooperate on numerous scientific projects, such as research and conferences. This network can be described as a virtual organization [21]. Virtual organizations can significantly profit from information technology support [29] and have already been confronted with SW [18].

The first stage of our hypothetical life cycle, involving the funding of research in the area of SW, can be regarded as accomplished in this case. It has resulted in numerous open source tools, such as Ontoviews [25], Sesame [6] or Protégé. These tools can now be utilized in the transition to the following stage.

The second stage should result in producing internal semantic data as well as internal added value. We have already mentioned Knowledge Management as a promising application field. Therefore, we designed, implemented and deployed a Skills Management system called TeamBuS. Our motivation was to test, whether such system brings added value and if so, how it influences the possibility of a transition to the following stage.

TeamBuS provides user with three basic functions. It enables managers creating teams to search after persons who best match the requirements of a given project. Here, the skills and experience play the major role. Moreover, it helps in finding experts in a particular area. Except for skills and experience, the accessibility of persons is very important. We provide information about most popular contact methods, including email, phone, instant messengers and internet telephony. Lastly, TeamBuS enables managers to analyse an organization's knowledge base. It does so in providing the capability of browsing through skills ontology and showing, how many persons of a particular organization or department fall into particular categories. The architecture of the system is shown in Fig. 3.

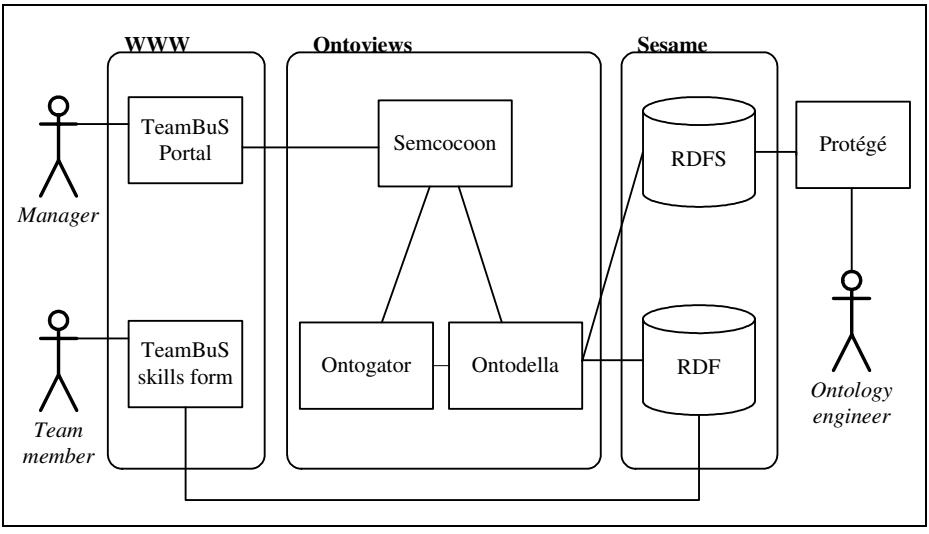


Fig. 3. TeamBuS architecture

The prototype version of the system has been tested in an organization gathering students interested in computer science. There were 35 users registered as team members in the system, and 13 users in the role of a manager, who tested the system and completed our evaluation survey. While this does not allow statistically significant conclusions, general trends can already be assessed.

The survey was based on scenarios, which provided test users with tasks compatible with the goals of the system. After accomplishing these tasks, they had to complete a survey containing 33 questions dealing with the usefullness of the system, our SW life cycle hypothesis, the performance of the system and users' profile.

The functionality of TeamBuS has proven very successfull – its usefullness in comparison with traditional keyword-based search has been graded an average of 4,4 (variance 0,3) on a 5 point scale. Our conclusion is that the system fulfills the goals prescribed in the second stage of our hypothetical life cycle. Firstly, it produces semantically annotated data. Secondly, it brings added value to the users. These two conditions present a requirement for the transition to the third stage of our model.

The third stage includes the scenarios B2B and negotiations, which present a transition step in bringing the SW to its full potential. In the survey, we asked the users how they perceived the potential of the system to be part of such scenarios. One possibility was the extension on the whole virtual organization, which would enable forming of cross-functional teams for the realisation of complex projects. Other possibility was to expose TeamBuS data as Semantic Web Services. These could be approached by agents of commercial firms in the recruitment process, negotiating the art and conditions of cooperation.

The users welcomed the extension of the system on other organizations with an average of 4,3 (variance 1) on a 5 point scale. They were motivated first of all by the

possibility of finding experts in other organizations – 4,7 (0,4), followed by creating interorganizational teams – 4,3 (0,6). The analytical tasks of an extended system were least appreciated – 4,1 (1,2). In the scenario involving recruiting companies, plain exposure of the users' data was not well accepted – 3 (1,5). However, when given the possibility to influence the communication process, the grade goes up – 4,4 (1,3). We conclude that there is need for negotiating software agents, if more advanced integration scenarios are to be accepted. Just plain semantic annotation of data is not enough, as organizations are not willing to give their data away for free. The users stressed that security would be an important factor in such scenarios.

The conclusion from this part of the survey is, that also in the third stage of the model added value can be realized. It would further increase the amount of semantic data and enable the transition to the last, fourth stage. However, a separate proof of concept for this stage including application and evaluation should also be conducted.

7 Conclusions

Our study aimed at developing an assessment framework for SW applications. Based on this framework we have formulated a hypothesis regarding the life cycle of these applications in organizations. As a proof of concept, we have designed, deployed and evaluated a Skills Management system.

The evaluation of the system strengthened our hypothesis, that the following life cycle phase (B2B, negotiations) is likely to happen. The next steps would be to extend the system on the whole virtual organization and design user agents, which will be able to negotiate with agents from recruiting companies. From this point, the enabling potential of the mass of data should be evaluated. We expect tighter cooperation between organizations after adopting TeamBuS.

As the system has been developed completely with open source tools, this experience might move small enterprises to adopt SW technologies. We have not experienced significant overhead in comparison with traditional COTS tools, and the system does provide added value.

References

1. V. R. Benjamins, J. M. L. Cobo, J. Contreras, J. Casillas, J. Blasco, B. de Otto, J. García, M. Blázquez, J. M. Dodero, "Skills Management in Knowledge-Intensive Organizations", *Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference*, EKAW 2002, Siguenza, Spain, Octover 1-4, 2002, Proceedings, pp. 80-95
2. V. R. Benjamins, D. Fensel, A. Gómez-Pérez, "Knowledge Management through Ontologies", Practical Aspects of Knowledge Management, *Proceedings of the Second International Conference*, Basel, Switzerland, October 29-30, 1998
3. B. Berendt, A. Hotho, D. Mladenic, M. Someren, M. Spiopoulou, G. Stumme, "Web Mining: From Web to Semantic Web", *First European Web Mining Forum*, Springer, Berlin et al., 2004
4. T. Berners-Lee, J. Hendler, O. Lassila, "The Semantic Web", *Scientific American*, no. 284, 2001, pp. 34-43

5. Z. Bjelogrlic, D.-W. van Gulik, A. Reggiori, "Making Business Sense of the Semantic Web", *International Semantic Web Conference 2003*, Springer, Berlin et al., 2003, pp. 818-833
6. J. Broekstra, A. Kampman, F. van Harmelen, "Sesame: An Architecture for Storin gand Querying RDF Data and Schema Information*", Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, MIT Press, 2003, pp. 197-222
7. C. Bussler, "The Role of Semantic Web Technology in Enterprise Application Integration", *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, no. 26, 2003, pp. 62-68
8. P. Castells, B. Foncillas, R. Lara, M. Rico, J. L. Alonso, "Semantic Web Technologies for Economic and Financial Information Management", *ESWS 2004*, Springer, Berlin et al., 2004, pp. 473-487
9. J. Contreras, V. R. Benjamins, M. Blázquez, S. Losada, R. Salla, J. Sevilla, D. Navarro, J. Casillas, A. Mompó, D. Patón, L. Rodrigo, P. Tena, I. Martos, "International Affairs Portal: A Semantic Web Application", *ECAI Workshop on Application of Semantic Web Technologies to Web Communities 2004*
10. J. Davies, A. Duke, Y. Sure, "OntoShare - An Ontology-based Knowledge Sharing System for virtual Communities of Practice", *The Journal of Universal Computer Science*, vol. 10, no. 3, 2004, pp. 262-283
11. J. Davies, D. Fensel, F. Harmelen, *Towards the Semantic Web*, John Wiley & Sons, Chichester, 2003
12. Y. Ding, D. Fensel, M. C. A. Klein, B. Omelayenko, "The semantic web: yet another hip?", *Data & Knowledge Engineering*, vol. 41, no. 2-3, 2002, pp. 205-227
13. J. Domingue, M. Martins, J. Tan, A. Stutt, H. Pertusson, "Alice: Assisting Online Shoppers through Ontologies and Novel Interface Metaphors", *Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, Springer, Berlin et al., 2002, pp. 335-351
14. D. Fensel, *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*, Springer, Berlin, 2001
15. D. Fensel, C. Bussler, Y. Ding, V. Kartseva, M. Klein, M. Korotkiy, B. Omelayenko, R. Siebes, "Semantic Web Application Areas", *Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems*, Stockholm - Sweden, June 27-28, 2002
16. S. Haustein, J. Pleumann, "Is Participation in the Semantic Web Too Difficult?" *International Semantic Web Conference 2002*, Springer, Berlin et al., 2002, pp. 448-453
17. O. Ibidunni, "Supporting workgroups collaborating via email using the semantic web and RDF", Hewlett-Packard, 2002
18. V. Iosif, P. Mika et al., "On-To-Knowledge: EnerSearch Virtual Organization. Case Study: Evaluation Document", *On-To-Knowledge Deliverable 29*, 2002
19. Y. Kalfoglou, H. Alani, W. M. Schorlemmer, C. Walton, "On the Emergent Semantic Web and Overlooked Issues", *International Semantic Web Conference 2004*, Springer, Berlin et al., 2004, pp. 576-590
20. D. Kang, B. Xu, J. Lu, Y. Zhang, "CSCW in Design on the Semantic Web", *Grid and Cooperative Computing: Second International Workshop*, Springer, Berlin et al., 2004
21. E. C. Kasper-Fuehrer, N. M. Ashkanasy, "The Interorganizational Virtual Organization", *International Studies of Management & Organization*, vol. 33, no. 4, 2003-4, pp. 34-64
22. T. Lau, Y. Sure, "Introducing Ontology-based Skills Management at a Large Insurance Company", *Modellierung in der Praxis - Modellierung für die Praxis, Arbeitstagung der GI, 25.-27. März 2002 in Tutzing, Proceedings*, pp. 123-134
23. H. Lausen, M. Stollberg, R. L. Hernández, Y. Ding, S.-K. Han, D. Fensel, "Semantic Web Portals – state of the art survey", Technical Report, DERI, 2004

24. A. Maedche, S. Staab, N. Stojanovic, R. Studer, Y. Sure, "SEmantic portAL: The SEAL Approach", *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, MIT Press, 2003, pp. 317-359
25. E. Mäkelä, E. Hyvönen, S. Saarela, K. Viljanen, "OntoViews - A Tool for Creating Semantic Web Portals", *International Semantic Web Conference 2004*, Springer, Berlin et al., 2004, pp. 797-811
26. S. Mantel, M. Schissler, "Application Integration", *Wirtschaftsinformatik*, no. 44, 2002, pp. 171-174
27. M. Montebello, "DAML enabled Agents and Workflow Components Integration, *Proceedings of the IADIS International Conference WWW/Internet 2002*, Lisbon, 2002
28. L. Nixon, M. Mochol et al., "KnowledgeWeb Deliverable D1.1.2 Prototypical Business Use Cases", Technical Report, NoE Knowledge Web, 2004
29. D. E. O'Leary, D. Kuokka, R. Plant, "Artificial Intelligence and Virtual Organizations", *Communications of the ACM*, vol. 40, no. 1, 1997, pp. 52-59
30. R. Schmaltz, "Semantic Web Technologien für das Wissensmanagement", Institut für Wirtschaftsinformatik, Georg-August-Universität Göttingen, Göttingen, 2004
31. R. Schmaltz, S. Hagenhoff, "On the Suitability of Semantic Web Approaches for Corporate Knowledge Management", *Proceedings of the VIIth SAM/IFSAM World Congress*, Göteborg, 2004
32. M. C. Schraefel, N. R. Shadbolt, N. Gibbins, S. Harris, H. Glaser, "CS AKTive space: representing computer science in the semantic web", *Proceedings of the 13th international Conference on World Wide Web WWW '04*, ACM Press, New York, 2004, pp. 384-392
33. Y. Sure, A. Gómez-Pérez, W. Daelemans, M.-L. Reinberger, N. Guarino, N. Fridman Noy, "Why Evaluate Ontology Technologies? Because It Works!", *IEEE Intelligent Systems*, vol. 19, no. 4, 2004, pp. 74-81
34. Y. Sure, V. Iosif, "First Results of a Semantic Web Technologies Evaluation", *Proceedings of the Common Industry Program at the federated event co-locating the three international conferences: DOA/ODBASE/CoopIS'02*, University of California, Irvine, 2002, pp. 69-78
35. S. Toivonen, T. Pitkäranta, J.U. Min, "On Describing B2B Processes with Semantic Web Technologies", *ERCIM News*, no. 56, 2003
36. Y. Zhao, K. Sandahl, "Potential Advantages of Semantic Web for Internet Commerce", *Proceedings of the 5th International Conference on Enterprise Information Systems,* Angers, France, April 22-26, 2003, pp. 151-160

Ontology-Based Searching Framework for Digital Shapes

Riccardo Albertoni[1], Laura Papaleo[2], Marios Pitikakis[3], Francesco Robbiano[1], Michela Spagnuolo[1], and George Vasilakis[3]

[1] IMATI-CNR, Via De Marini, 6, Torre di Francia, 16149 Genova, Italy
{albertoni, robbiano, spagnuolo}@ge.imati.cnr.it
[2] Department of Informatics and Computer Science, University of Genova,
Via Dodecaneso, 35 - 16100 Genova, Italy
papaleo@disi.unige.it
[3] Center for Resaerch and Technology Hellas, Informatics and Telematics Institute,
1st Km Thermi-Panorama Road, 57001 Thermi-Thessaloniki, Greece
{pitikak, vasilak}@iti.gr

Abstract. Knowledge related to Shape Modelling is multi-faceted because of the complexity and heterogeneity of the involved resources and because different applications may cast different semantics on them. A fast evolution of the field is now conditioned by how research teams will be able to communicate and share resources and knowledge. The field needs to be formalized in order to achieve a shared conceptualization accessible by the whole scientific community and eventually to ensure an actual exploitation of its knowledge within the Semantic Web. In this context, the main objective of the Network of Excellence AIM@SHAPE is twofold: on the one hand to devise tools to capture the implicit semantics of digital shapes, and on the other hand to encode and formalize the domain knowledge into context-dependent ontologies. The paper describes the first results in the direction of developing an ontology for shape acquisition and reconstruction and its effective use in the Digital Shape Workbench, a searching framework for sharing resources (shapes, tools and publications) and their related knowledge.

1 Introduction

The success of the scientific enterprise largely depends on the ability of sharing scientific resources (information, papers, tools) among the scientific community. In the last decade the web has been emerging as a mean to fulfil this requirement by facilitating the communication and by making easily available a huge amount of information. This problem is particularly relevant in the field of Shape Modelling, which concerns methods to represent, create, process and analyse digital representations of objects for a variety of applications. The most typical kind of resources in this field are *digital shapes*, i.e. multi-dimensional media characterized by a visual appearance in a space of 2, 3, or more dimensions. Examples of shapes are pictures, images, 3D models, videos (disregarding the sound track), animations, etc.

Shape Modelling includes Computer Graphics and Vision and it is based on a large spectrum of fundamental domains, including differential geometry, numerical analysis, computational geometry and discrete topology. Recently, the field has reached a state where each individual fundamental domain is well understood and exploited. A

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 896 – 905, 2005.
© Springer-Verlag Berlin Heidelberg 2005

fast evolution of the field is now conditioned by how research teams will be able to intercommunicate. Beside the need of an *e*-science platform for supporting research in the field, shapes are gaining importance in different social contexts. Considering that most PCs connected to the Internet are now equipped with high-performance 3D graphics hardware, it seems clear that in the near future 3D data will represent a huge amount of traffic and data stored in the Internet. It has been predicted that geometry is poised to become the *fourth wave* of digital multimedia communication, where the first three waves were sounds in the 70's, images in the 80's, and videos in the 90's. Digital shapes are therefore expected to take a central role in the Semantic Web in the coming years, with high potential impact in several key areas.

In this context, the Network of Excellence AIM@SHAPE [1] is pursuing the introduction of Knowledge Management techniques in Shape Modelling, with the aim of making explicit and sharable the knowledge embedded in digital shapes. On the one hand, this requires the development of tools able to extract semantics from 3D models (e.g. automatic or semi-automatic annotation tools), on the other hand it is necessary to build a common framework for reasoning, searching and interacting with the semantic content related to the knowledge domain. As pointed out by Hendler [2] researchers may need to find and explore results at different levels of granularity, from other perspectives in the field or from a complete different scientific field. Although scientists are relying on the web to share their own scientific resources, the current Web technology is clearly insufficient for the need of supporting collaborative e-science. In AIM@SHAPE the Digital Shape Workbench (DSW, for short) is a more elaborated framework to store shapes, tools, publications along with the knowledge related to them, relying on a search engine able to provide significant results. The development of the DSW for the complex field of Shape Modelling requires the conceptualisation of the domains and the precise characterization of the resources. The AIM@SHAPE effort can be seen as a step towards contributing to the goal of the Semantic Web itself. As a matter of fact, the success of the Semantic Web as the mean to share scientific resources is significantly limited, if a shared conceptualisation of scientific fields will not emerge.

The paper aims at presenting the contribution of AIM@SHAPE to the harmonization of content in the field of Shape Modelling. The complexity and the wideness of the domain makes unreasonable to provide a shared conceptualisation in terms of one monolithic ontology, and it forces in building a framework where different ontologies are adopted to represent facets of specific domain applications and usage scenarios. In particular, the paper presents fragments of an ontology which formalises the knowledge related to the pipeline of Acquisition and Reconstruction of digital shapes. The paper will briefly review the status of the tools needed to build a semantic-based platform for Shape Modelling. Then, the AIM@SHAPE approach of modelling the semantics of digital shapes and shape resources will be introduced and a detailed description of the acquisition phase of a shape will be given. Finally, the DSW search architecture will be briefly sketched, and conclusions will be drawn.

2 Related Works

While academic and research communities have historically been key contributors to the development of the Internet, the potential of Internet as a tool for collaborative

research activity have been only recently understood. In the field of Shape Modeling, the use of Internet as a mean for collaborative environment has been mainly focused on setting up benchmarking for testing the performance of different algorithms. The most famous example is the Stanford Repository [3], a collection of downloadable models obtained by scanning, documented by rather simple attributes. The site is a simple HTML page, with limited search capabilities.

The retrieval of digital shapes in large heterogeneous repositories is still a complex task. Information encoded in multi-dimensional media, unlike text data, is totally *implicit*, being based on data formats that have no relation with data interpretation and offer no grasp to their direct access and easy understanding. Browsing, retrieving and navigating efficiently in video or image databases is not easy at all, not to mention databases of 3D shapes and data volumes. At the state-of-the-art, the only effective means to perform context-based retrieval on such databases rely on textual annotations of media (e.g. keywords), which are inserted manually and constitute only a negligible portion of the information stored in the repository. In the last years, there has been quite a lot of effort in the Shape Modelling community for providing smart tools able to retrieve three-dimensional data using shape matching [3]. These engines address the problem in a geometric sense, so they are able, at a certain extent, to retrieve objects that present some geometric similarity. The peculiarities of the field make the general problem of retrieving intrinsically complex. The knowledge is not solely carried by digital shapes, but also by hardware and software tools used to acquire and transform them. Moreover, shapes are heterogeneous, as they can be represented in different ways with regard to both format and content. Being multi-dimensional data, the size of digital shapes is generally very big: for accurate 3D models, the size can be some GigaBytes each. Last but not least, shapes are used in different environments such as: Industrial Design (e.g., CAD models of products,), Cultural Heritage, Medical Applications (e.g., tomography), Entertainment (e.g., computer animations), Geographical Information Systems (e.g., three-dimensional models of terrains), and many more.

The support of querying facilities has always been a primary requirement for repositories of any kind. Of course, the simplest approach is to search for keywords in filenames, captions, or context. However, this approach is highly inefficient. Moreover, the current digital shapes repositories are centered on the geometric aspect of shapes, and not on the knowledge they represent. Different methods for measuring similarity between shapes have been presented [4], [5]. Content-based retrieval and classification systems have also been developed for other multimedia data types, including audio [6], images [7], and video [8]. The representation of a shape can be sorted according to three levels of sophistication: the Geometric level, the Structural level and the Semantic level. While on the geometric and structural level there are numerous approaches, at the semantic level very little work has been done until now. In the last few years, apart from the AIM@SHAPE Network of Excellence [1], there has been a considerable increase of interest for techniques to extract and stream knowledge embedded into multimedia content, ranging from basic research efforts to projects seeking an integrated effort at European level [10], [11].

The proliferation of knowledge caused by the widespread use of the Web as a knowledge communication platform has posed the same and even more imperative requirements for performing queries and locating resources into the vast information space. We believe that the addition of explicit semantics can improve search. How-

ever, the data models used to represent and encode knowledge on the Web differ from the traditional data structures. RDF [12], RDFS [13] and OWL [14] are the emerging standards used to encode web-based data. Thus, the functionality a querying language should support the structure and the peculiarities of the new paradigms. Some query languages have been developed for RDF/S (e.g. RQL [15], SquishQL[16], TRIPLE[17]), and DAML (e.g. DQL [18], RDQL[19]). For querying OWL semantic web repositories, the query language OWL-QL [20] has been proposed, which is the successor of the DQL query language and takes advantage of the expressive power of the OWL language itself. OWL-QL is a language with precisely defined semantic relationships among a query, a query answer, and the knowledge base(s) used to produce the answer. Practical description logic (DL) systems such as Racer [21] offer a functional API for querying a knowledge base. The first step towards satisfying more expressive querying facilities provides the new Racer Query Language (nRQL) [22].

3 Ontology-Driven Annotation of Shapes: The AIM@SHAPE Approach

Due to the intrinsic complexity of shapes, ontology-driven metadata are necessary in order to reach a sufficient level of expressiveness. Metadata should provide a thorough characterization of shapes (**Fig.1**) by storing: *(i)* the information related to its history, such as the acquisition devices and techniques for creating it or the tools for transforming it (its *past*, e.g. for documentation), *(ii)* the information intrinsically held by the shape itself (its *present*) and *(iii)* the information related to its capabilities and potential uses, such as the possible steps that can be performed or the tools that can be used (its *future*, e.g., for acquisition/process planning).

Moreover, ontology-driven metadata should be able to represent different levels of sophistication describing a shape as a *simple resource* (e.g. for cataloguing) and characterizing it according to its *geometry* (e.g. for rendering), to its *structure* (e.g. for matching and similarity), and to what it *represents* (e.g. for recognition or classification). **Fig. 2** gives an example of a digital shape and its intrinsic characteristics: it can be seen as simple resource (e.g. name and URL), or can be considered by its geometric characteristics (e.g. a set of triangles and normals). It has a structure (e.g. the skeleton of a teapot) or it can be seen a teapot composed by a handle, a spout, a body and a tip. It is important also to take into account the different environments where the shape can be used since the specific application determines relevant characteristics. For example, if the main purpose is to build a teapot, the identification of parts by which a teapot is composed is fundamental, while if the purpose is to let a robot grasp it, the localization of the handle is the only necessary task.

The existing branches of research in the field of Shape Modelling (e.g. Computer Graphics and Vision) are interested in one or more of the above mentioned characterizations, but also on the conditions and the tools to pass from one characterization to another. Notice that shapes play a central role in Shape Modelling, but they do not represent the only kind of resource that must be characterized in the common framework.

Everyday, scientists work with shapes, tools and publications. It is important to devise the role of these resources in the different conceptualizations, making relation ships among them explicit. For example, a scientist may want to evaluate his latest

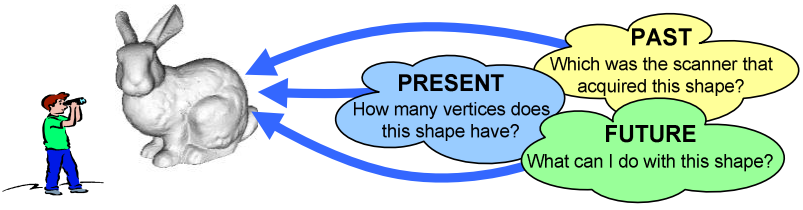


Fig. 1. An expressive characterization of a shape is made up by the information related to its history, the information intrinsically held by the shape itself and the information related to its capabilities

implementation of a method. In this case, it is interesting to figure out which are the tools providing other implementations of the same method, the publications related to the above tools and methods, the shapes used as tests for the other implementations (e.g. for testing/benchmarking activities). It was decided to enhance the semantic aspects of shapes using two different and coexisting approaches [9]. The first strategy is *analytic* and acts on the side of Shape Modelling: development of tools and methods to extract morphological structures from low-level geometry (e.g.: find the skeleton of a shape), and semantic information from structures (e.g. trying to understand where is the handle of a door or the tip of a teapot). The second strategy is *synthetic*, and acts on the side of Knowledge Technologies: the domain knowledge and the shape semantics are encoded in context-dependent ontologies, and are used, for example, to annotate and retrieve shapes.

Concerning the synthetic strategy, three main ontologies have been initially addressed within AIM@SHAPE (Virtual Humans [23], Product Design [24] and Acquisition and Reconstruction of shapes [25]). These ontologies are used in the DSW to browse the collected resources according to context-dependent views (Section 5).

To give a flavour of what is meant by conceptualising one of the mentioned application domains, the next section describes the Acquisition and Reconstruction ontology. In particular, the fragment related to acquisition of a real object will be detailed.

4 An Ontology for Shape Acquisition and Reconstruction

The design of the ontology for Shape Acquisition and Reconstruction (AR, for short) follows mainly the *On-To-Knowledge* methodology [26] which is characterized by the specification of the requirements and an iteration of refinement, evaluation and maintenance phases. The domain of the ontology has been defined as the development, usage and sharing of hardware tools, software tools and shape data by researchers and experts in the field of acquisition and reconstruction of shapes. To specify the AR pipeline the following macro-steps have been defined: (1) *Shape Acquisition* (and Registration): the phase in which sensors capture measurements from a real object; (2) *Shaping*: the phase in which all acquired data are merged to construct a single shape; (3) *Shape Processing*: the phase in which further computations on the shape may be done (e.g. smoothing, simplification, enhancement, and so on).

As we said before, the AR ontology is intended to be targeted to the scientific community. For this reason, within AIM@SHAPE, experts of the field were inter

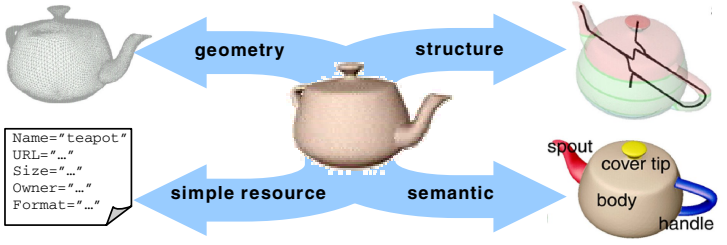


Fig. 2. A shape is described as a simple resource, or by its geometry, its structure, its semantics, depending on the application domain

viewed to understand the requirements and to sketch the competency questions. From the feedback obtained, it was clear that an important landmark of this ontology would have been the conceptualisation of the *Acquisition Session*, with the main aims of planning the acquisition of real objects and of annotating the shapes by documenting their acquisition.

Besides, it is important to remind that a proper conceptualization of shapes, tools, and publications is fundamental not only for their own characterization but also to provide meaningful cross-correlations. Nevertheless, in the next subsection we will mainly focus on the Acquisition Session, which represents a fragment of the overall ontology, as a significant example to demonstrate why our ontology can be used for gathering the resources and the related knowledge.

4.1 Modelling the Knowledge of Shape Acquisition

The *AcquisitionSession* has been modelled as a concept in the AR ontology. It is related to an *AcquisitionSystem* (which is made up by one or more *AcquisitionDevices,* e.g. scanners) and to the conditions in which the acquisition is performed: the *LogisticConditions* (they include the presence of lights, if there exist any obstacle between the real object and the scanning device and so on) and the *EnvironmentConditions* (which include the information on where the real object is –indoor or outdoor or underwater- or the level of humidity or even the weather). Moreover, some attributes are directly related to the *AcquisitionSession* (e.g. the price for renting the technological devices), while other are related to the different entities in the framework (e.g. the price of a scanning system, or the person/institute responsible for it). An overview on the conceptualisation of the Acquisition Session is given in **Fig. 3** where each rectangle represents a concept. The rows in each concept represent a slot which can be either an attribute or a relationship. For each attribute the type is specified, while for each relationship it is indicated the range. Whenever a symbol '*'appears close to the name of an attribute or a relationship, the multiplicity can be more than 1.

An *AcquisitionSession* basically documents the acquisition of a *RealObject* and the production of a *ShapeData* (a digital shape), using a particular *AcquisitionSystem. ShapeData* has been also modelled as a concept in our ontology, with some properties, such as its format or its URL, but also the information on the source from which it has been generated (through the slot *hasSource*). Taken the ontology fragment related to *AcquisitionSession* and *ShapeData* as an example, it can be shown that our ontology is

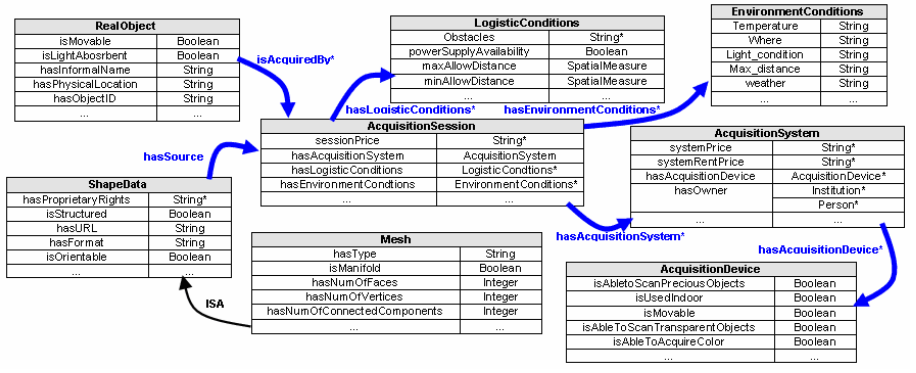


Fig. 3. A fragment focused on *AcquisitionSession* in our ontology for Shape Acquisition and Reconstruction. The most significant relationships are highlighted by arrows

able to support in obtaining the knowledge associated to a digital shape, such as the description of what we called its past, its present and its future. For example, an instance of *AcquisitionSession* includes information about the scanner used to acquire a real object constituting important documentation about *ShapeData*'s past. Supposing that the type of the produced Shape Data is a *Mesh,* we can focus on some information intrinsically held by itself (and so related to its present)*, e.g.* the number of vertices or faces.

At the same time, the number of vertices of a *Mesh* can be useful to plan future steps. For example, if we are interested in a surface mesh with at least 10.000 vertices and we found a surface mesh with 3.000 vertices, we could decide to plan a new *AcquisitionSession* increasing the accuracy of the *AcquisitionSystem*. In this case, the ontology supports the planning of a new *AcquisitionSession* providing information such as which *AcquisitionSystem*s are available (indicating also the owners of them), the prices to rent these systems, and so on. The concepts represented in the ontology, being selected according to the experts' skills, provide the right expressiveness to describe and to gather the resources.

5 The DSW Architecture to Support Search

The *Digital Shape Workbench* (DSW) is aimed at laying the basis for a common research platform for modelling, storing, processing and reasoning about shape models and software tools. At the core of the infrastructure, the ontology and metadata server constitutes the knowledge base that conceptualizes and provides persistency services for the knowledge in the field of shapes. Built on top of the ontology server, services for supporting inference and searching are provided.

The shape models and the software tools are organized in distributed repositories accessible via common APIs. A high-level view of the architecture is shown in **Fig.4**. The Search and Inference Engine is probably the most important component of the DSW architecture and addresses the problem of searching for available resources in the knowledge base. The purpose of this engine is to provide non-trivial quality of

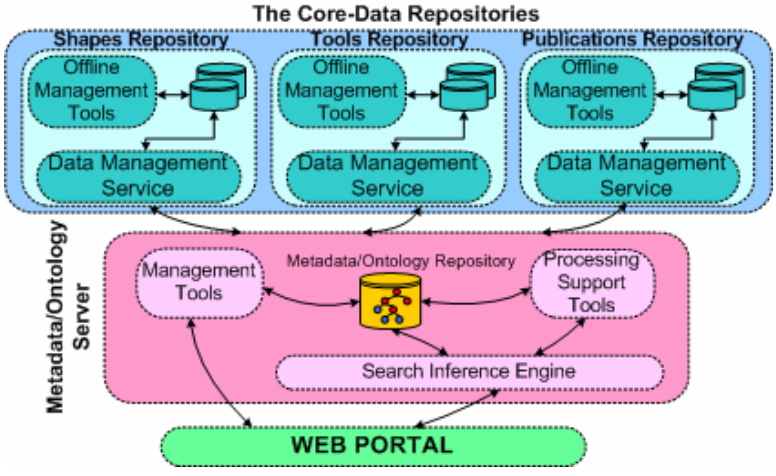


Fig. 4. The DSW Architecture. The core data repositories are interrelated with the Metadata/ontology server and queried from the web portal via the Search Inference Engine.

service. In this case we are not simply interested in searching for resources; rather we are interested in searching intelligently.

We deal with this objective by specifying the ontologies (as the AR ontology presented in the previous section) in OWL, which provides some support for deductive reasoning, and by using a DL reasoner as Racer for providing the required inference capabilities.

The search engine provides a unified interface, to be used for accessing metadata information stored in the domain ontologies. Queries submitted using the search engine interface will place some semantic criteria on the metadata associated with digital shapes. The search engine will then use deductive reasoning and inference to find resources that match the specified criteria. One of the most important aspects of searching is to establish how to search. In our case, the way of searching is related to the user comprehension of the domain and of the structure of its conceptualization.

Anyway, the AR ontology has been built taking into account the knowledge of the experts in the field of Shape Modelling. This ensures that it provides the right expressiveness to describe and to gather the resources (shapes, tools and publications).

To help the user in making efficient and appropriate queries, taking full advantage of the search and inference mechanisms, we are developing a graphical user interface that adds yet another level of abstraction. To simplify GUI development, a semantic layer is built on top of Racer that uses the DIG interface [27] for communication, thus ensuring independence of reasoner specific functionality. This layer provides basic class- and instance-level reasoning constituting a general-purpose framework for accessing inference engines that support the DIG language. The goal of the graphical interface is to provide the user with the means to search in an intuitive and straightforward way, without sacrificing flexibility and expressiveness of the queries. Furthermore, the user is able to store and reuse predefined or user-defined queries. Processing support tools provide additional functionality in answering queries that the ontology mechanisms alone cannot answer. This kind of queries does not involve only domain knowledge, which is captured by the ontology, but some processing as well. Examples of such queries are: trans-

forming a shape from one representation to another, producing some similarity estimation between two shapes, and so on. Management tools are provided in the DSW in order to assist in the efficient management of the ontology and metadata repository. These include tools for creating, editing, parsing and validating, loading, browsing and visualizing ontologies and metadata descriptions. Furthermore, a unified web interface to these tools will be provided, in order to, along with the search engine, provide a single point of access to ontology management operations.

The DSW constitutes the first step in the development of a large-scale e-science framework promoting research on shapes, by formalizing, processing and sharing knowledge about digital shapes and their applications. The scalability of this approach will eventually lead to the actual exploitation of the Shape Modelling domain knowledge within the Semantic Web.

6 Concluding Remarks

The paper proposes an ontology-based searching framework for digital shapes. It aims to address the need of a new approach to store and retrieve shapes, tools and publications related to the field of Shape Modelling. This need is rapidly emerging from different social contexts and in particular from the scientific community. The proposed framework relies on the Digital Shape Workbench (DSW) and on a conceptualisation of the domains within the field of Shape Modelling. The DSW provides a common research platform for modelling, storing, processing and reasoning about digital resources, whereas the conceptualisation provides a characterization of the relevant resources and their related knowledge in order to retrieve them with a sufficient expressiveness. Due to the complexity of the field, it is not possible to represent the conceptualisation in terms of a monolithic ontology and therefore different ontologies have been designed. The aim is to address multiple contexts and applications where the shape knowledge can be exploited. In particular, the paper presents the DSW architecture and the ontology for Shape Acquisition and Reconstruction, as a portion of the whole conceptualisation, in order to demonstrate the capabilities of the entire framework.

The contribution of this work is twofold: on the one hand it contributes to the goal of the Semantic Web, adding essential semantics for content-based information and knowledge retrieval; on the other hand it boosts the scientific enterprise paving the way to a more efficient collaboration among scientists.

Acknowledgments

This work has been supported by the ECFP6 IST NoE 506766 AIM@SHAPE. Special thanks to all the partners in the AIM@SHAPE project and by insights in the field, Marco Attene, Simone Marini, Giuseppe Patané, Alfonso Quarati, Paolo Cignoni and Rita Borgo of IMATI-CNR and Francesca Odone of DISI, University of Genova.

References

1. ECFP6 IST NoE 506766 AIM@SHAPE Network of Excellence, www.aimatshape.net
2. Hendler, J.: Science and the Semantic Web. Science, Vol.299, 5606, (2003), 520-521
3. The Stanford 3D Scanning Repository, http://graphics.stanford.edu/data/3Dscanrep/

4. Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., Jakobs,D.: A Search Engine for 3D Models. ACM Transactions on Graphics, Vol.22, 1, (2003)83-105
5. Daras, P., Zarpalas, D., Tzovaras, D. Strintzis, M.G.: Shape Matching Using the 3D Radon Transform. 3D Data Processing, Visualization & Transmission, Thessaloniki, (2004)
6. Foote, J: An overview of audio information retrieval. ACM Multimedia Systems, Vol.7 (1999)2–10
7. Castelli,V. Bergman, L.:Image Databases: Search and Retrieval of Digital Imagery. John Wiley & Sons (2001)
8. Veltkamp, R.C., Burkhardt, H., Kriegel H.P. (eds.): State-of-the-Art in Content-Based Image and Video Retrieval. Computational Imaging and Vision, Vol. 22, Kluwer Acad. Publ. (2001)
9. Falcidieno, B., Spagnuolo, M.,Alliez, P., Quak, E., Vavalis, M. and Houstis. C.: Towards the Semantics of Digital Shapes: The AIM@SHAPE Approach. In: European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology, London, UK, (2004)
10. SCHEMA NoE. www.schema-ist.org/SCHEMA
11. AceMedia Integrated Project, www.acemedia.org/aceMedia
12. Lassila, O., Swick, R.: Resource Description Framework (RDF) Model and Syntax Specification, W3C Candidate Recommendation, (1999), www.w3.org/TR/REC-rdf-syntax
13. Brickley, D., Guha, R.V: Resource Description Framework Schema (RDF/S) Specification 1.0, W3C Recommendation, (2000) www.w3.org/TR/rdf-schema
14. McGuinness D, van Harmelen , F. (ed.): OWL Web Ontology Language Overview (2003) www.w3.org/TR/owl-features/
15. Karvounarakis, G., Alexaki, S., Christophides, V., Plexousakis, D., Scholl, M: RQL: a declarative query language for RDF. In: WWW, (2002)592-603
16. Miller, L., Seaborne, A., Reggiori, A: Three implementations of squishql, a simple RDF query language. In International Semantic Web Conference, (2002) 423-435
17. Sintek, M., Decker, S: TRIPLE-An RDF Query, Inference, and Transformation Language. In Proc. of the Deductive Databases and Knowledge Management Workshop, Japan, (2001)
18. Fikes, R., Hayes,P., Horrocks I., (ed): DAML Query Language (DQL) (2003) www.daml.org/2003/04/dql/
19. Seaborne A.: RDQL (2004) www.w3.org/Submission/2004/SUBM-RDQL-20040109/
20. Fikes R., Hayes, P., Horrocks, I.: OWL Query Language (OWL-QL) Abstract Specification, DARPA Agent Markup Language (DAML) (2003)
21. RACER, www.racer-systems.com
22. Haarslev V., Möller, R., Wessel, M.: Querying the Semantic Web with Racer + nRQL. In Proceedings of the Intern. Workshop on Applications of Description Logics, Germany, (2004)
23. Gutierrez, M., Thalmann, D., Vexo, F., Moccozet, L., Magnenat-Thalmann N., Mortara, M., Spagnuolo, M.: An Ontology of Virtual Humans: incorporating semantics into human shapes. In: Proceedings of workshop SVE05, Switzerland (2005)
24. Ucelli, G., Brunetti, G., De Amicis, R., Conti, G.: Shape Semantics and Content Management for Industrial Design and Virtual Styling. In: Proceedings of SVE05, Switzerland (2005)
25. Papaleo, L., Albertoni, R., Marini, S., Robbiano, F.: An ontology-based Approach to Acquisition and Reconstruction. In: Proceedings of workshop SVE05, Switzerland, (2005)
26. Sure Y., Staab, S., Studer, R.: On-To-Knowledge Methodology. In: S. Staab and R. Studer, (eds.): Handbook on Ontologies, series on Handbooks in Information Systems, Springer, (2003) 117-132
27. DIG interface, http://dig.sourceforge.net/

Ontology Metadata Vocabulary and Applications

Jens Hartmann[1], Raúl Palma[2], York Sure[1], M. Carmen Suárez-Figueroa[2], Peter Haase[1], Asunción Gómez-Pérez[2], and Rudi Studer[1]

[1] Institute AIFB, University of Karlsruhe, Germany
[2] Ontology Engineering Group, Laboratorio de Inteligencia Artificial, Facultad de Informática, Universidad Politécnica de Madrid, Spain

Abstract. Ontologies have seen quite an enormous development and application in many domains within the last years, especially in the context of the next web generation, the Semantic Web. Besides the work of countless researchers across the world, industry starts developing ontologies to support their daily operative business. Currently, most ontologies exist in pure form without any additional information, e.g. authorship information, such as provided by Dublin Core for text documents. This burden makes it difficult for academia and industry e.g. to identify, find and apply – basically meaning to reuse – ontologies effectively and efficiently. Our contribution consists of (i) a proposal for a metadata standard, so called Ontology Metadata Vocabulary (OMV) which is based on discussions in the EU IST thematic network of excellence Knowledge Web[1] and (ii) two complementary reference implementations which show the benefit of such a standard in decentralized and centralized scenarios, i.e. the Oyster P2P system and the Onthology metadata portal.

1 Introduction

Ontologies are commonly used for a shared means of communication between computers and between humans and computers. To reach this aim, ontologies should be represented, described, exchanged, shared and accessed based on open standards. Consider, as an example, the W3C standardized web ontology language OWL [10]. Currently, most ontologies exist in pure form without any additional information, e.g. authorship information, such as provided by Dublin Core for text documents. This burden makes it difficult for academia and industry to identify and apply – basically meaning to reuse – ontologies effectively and efficiently. Metadata is meant as machine processable information for the Web[2]. It is a systematic method for describing information resources, helps to improve their accessibility and gives other useful resource information to support their maintenance (e.g. to find data sets, to determine whether the data set is appropriate for a certain use, etc.). Thus, one key purpose of metadata is to facilitate and improve the retrieval of information.

Taking into account that ontology sharing and reuse is quite often difficult for academia and industry and the main features of metadata, they could be used for

[1] `http://knowledgeweb.semanticweb.org/`
[2] `http://www.w3.org/Metadata/`

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 906–915, 2005.
© Springer-Verlag Berlin Heidelberg 2005

describing ontologies (the outcome of this would be ontology metadata) for sharing, exchanging and reusing them in a most efficient way. To achieve this goal, it is necessary to agree on a standard for ontology metadata, that is a common set of terms and definitions describing ontologies, so called metadata vocabulary. Then, implementing such a vocabulary will increase the value of ontologies by facilitating ontology sharing and reusing through time and space. If ontologies are described using ontology metadata standards, an appropriate technology infrastructure is required. For example, tools and metadata repositories, compatible to the ontology metadata standards, must be developed. These tools and repositories can as a consequence e.g. support the creation, maintenance and distribution of ontology metadata.

Our contribution consists of (i) a proposal for a metadata standard for capturing properties ontologies supporting their reuse, so called Ontology Metadata Vocabulary (OMV), which is based on discussions and agreement in the EU IST thematic network of excellence Knowledge Web and (ii) two complementary reference implementations which show the benefit of such a standard in decentralized and centralized scenarios, i.e. the P2P system Oyster and the metadata portal Onthology. This paper is organized as follows: section 1 provides the introduction. The developed metadata vocabulary is given in section 2 introduced by the main requirements. The P2P system Oyster and the portal Onthology are described in section 3. In section 4 we provide related work, conclude and present future work.

2 Ontology Metadata Vocabulary

2.1 Requirements

As an initial step towards a standardized vocabulary, we analysed requirements for ontology metadata. Several aspects are similar to other metadata standards, like Dublin Core. However, important differences like the conceptual models (semantics) behind ontologies require a detailed analysis and require a different representation of metadata about ontologies. In a nutshell, an ontology normally reflects the (i) conceptualization from persons about a specific task or domain which then is (ii) realised by an ontology engineering process [12].

As a result, the main identified requirements are the following:

- **Accessibility:** Metadata, especially about ontologies, must be accessible and processable for humans and machines.
- **Usability:** a majority of users should be able to apply metadata easily.
- **Reuse of Ontologies:** As ontologies are a core technology for the Semantic Web, its metadata should reflect key issues of the Semantic Web as well. In particular **reuse** and **sharing** of knowledge.
- **Conceptualisation vs. Realisation:** Metadata must reflect (and also distinguish between) a semantic *conceptualisation* and its particular *realisation* as a concrete ontology document.
- **Interoperability:** Metadata should be interoperable and conform to the major representation languages currently being used for Semantic Web applications. Indeed, this means that a metadata vocabulary should be representable e.g. in F-Logic and OWL as well.

- **Documentary:** Documentary aspects of metadata like information about *technical, statistical, accessibility, management information, etc.* should be provided.
- **Extensibility:** Reflecting special user needs, it is required that beyond such standard metadata facts can be added and extended easily.
- **Expressiveness:** Metadata must be expressive enough to represent all desired aspects, as presented above.

The main aspects are *Conceptualisation vs. Realisation* and *Reuse of Ontologies* which should be reflected by any ontology metadata. Already now, it is possible to capture several technical properties of ontologies, like *used syntax* or *number of classes*, almost automatically like realised by [3] for example. Besides technical properties which are obviously relevant, there is a strong demand for representing conceptual metadata, like authorship information, categorizations or underlying methodologies. As consequence, representing these issues by a vocabulary requires an expressive language for the metadata itself which makes it impossible to reuse any existing metadata schema.

2.2 Conceptualisation vs. Realisation

OMV distinguishes between an **ontology base** and an **ontology document**. This separation is based on following observation: any existing ontology document has some kind of *core idea* (conceptualisation) behind. From an ontology engineering perspective, initially a person develops such *core idea* of what should be modeled (and maybe how) in his mind. Further, this initial conceptualisation might be discussed with other persons and after all, an ontology will be *realized* using an ontology editor and stored in a specific format. Over time, there might be created several *realizations* of this initial *cconceptualisation* in many different formats, e.g. in RDF(S) [1] or OWL [10].

Therefore we distinguish between an *ontology base* and an *ontology document*:

- **[Ontology Base]:** An *Ontology Base (OB)* represents the abstract or core idea of an ontology, so called conceptualisation. It describes the core properties of an ontology, independent from any implementation details. For a general illustration of the relationship of an OB and OD, we refer to figure 1.
- **[Ontology Document]:** An *Ontology Document (OD)* represents a specific realization of an ontology base. Therefore, it describes properties of an ontology that are related to the realization or implementation.

The distinction between an OB and OD leads to an efficient mechanism, e.g. for tracking several versions and evolvements of ontologies as well as for different representations of one knowledge model (conceptualisation) in different languages. In particular, such an *ontology base* can be seen as representation of the conceptual model



Fig. 1. Relationship between OB and OD

behind an ontology. Technically, an ontology base and an ontology document are modeled as two separate classes, with the relation `realizes` from the ontology document to the ontology base. This means that there may be many possible ontology documents for one ontology base, but one ontology document can only realize one ontology base.

Normally, an OD should not be able to exist without a corresponding OB. However, for practical reasons, we allow the existence of each class independently of each other. Hence, we cannot assume that every existing ontology will be annotated by its original author who might create an OB for his ontology. However, automatically extracting syntactical properties of an existing ontology is quite simple. Then, such *minimal OD* would exist without a concrete OB.

The main classes and properties of the OMV ontology are illustrated in figure 2. Please notice, that not all classes and properties are included. It is only to demonstrate the main idea behind OMV. The complete ontology is described in [6] and is available for download in several ontology formats[3].

It should be noticed that there exist several properties within OB and OD which look similar at first. However, they have different meanings and semantics. Exemplary, think of an ontology engineer `A` developing an ontology in RDF(S) syntax and annotating it with OMV. Then, the properties of an OB and OD individual are quite similar. Exemplary, both would have the same `party` as `creator` and so on. However, over time, there might be another engineer `B` with similar needs according to the OB from `A`. Hence, `B` reuses the OB from `A` and only creates a new OD, e.g. realizing the OB in OWL instead of RDF(S). As a result, a new OD would be created for this ontology and most of the properties are different now.

2.3 Properties to Support Reuse in OMV

As mentioned above, the OMV models the two main classes OB and OD for representing core information about ontologies. However, additional classes are required to represent and support the reuse of ontologies by such metadata vocabulary, especially in the context of the Semantic Web. Hence, we modeled, as shown in figure 2, further classes and properties representing *environmental information* and *relations*. We will briefly discuss these classes in the following. In typical ontology engineering mainly a `Person` (or multiple) or an `Organisation` as a whole are developing ontologies. We group these two classes under the generic class `Party` by a `subclass-of` relation. A `Party` can *create*, *contribute* and *review* an `OntologyBase` resp. an `OntologyDocument`. We here distinguish between the development of an OB and OD. Further, tools such as ontology editors can be referred to by the class `OntologyEngineeringTool` which itself can be `developedBy` a `Party`. The different existing syntactical representations and ontology languages are representable by `OntologySyntax` and `OntologyLanguage`. OMV further consists of the class `KM-Method` make explicit the methodology (or methodologies) used during engineering. Ontologies might be categorized by different types of ontologies, exemplary think of *domain, lingusitic* or *upper-level* ontologies. Those types can be modeled by the class `OntologyType`. For industry it might be relevant to propose usage licenses

[3] OMV representations are available at `http://ontoware.org/projects/omv/`

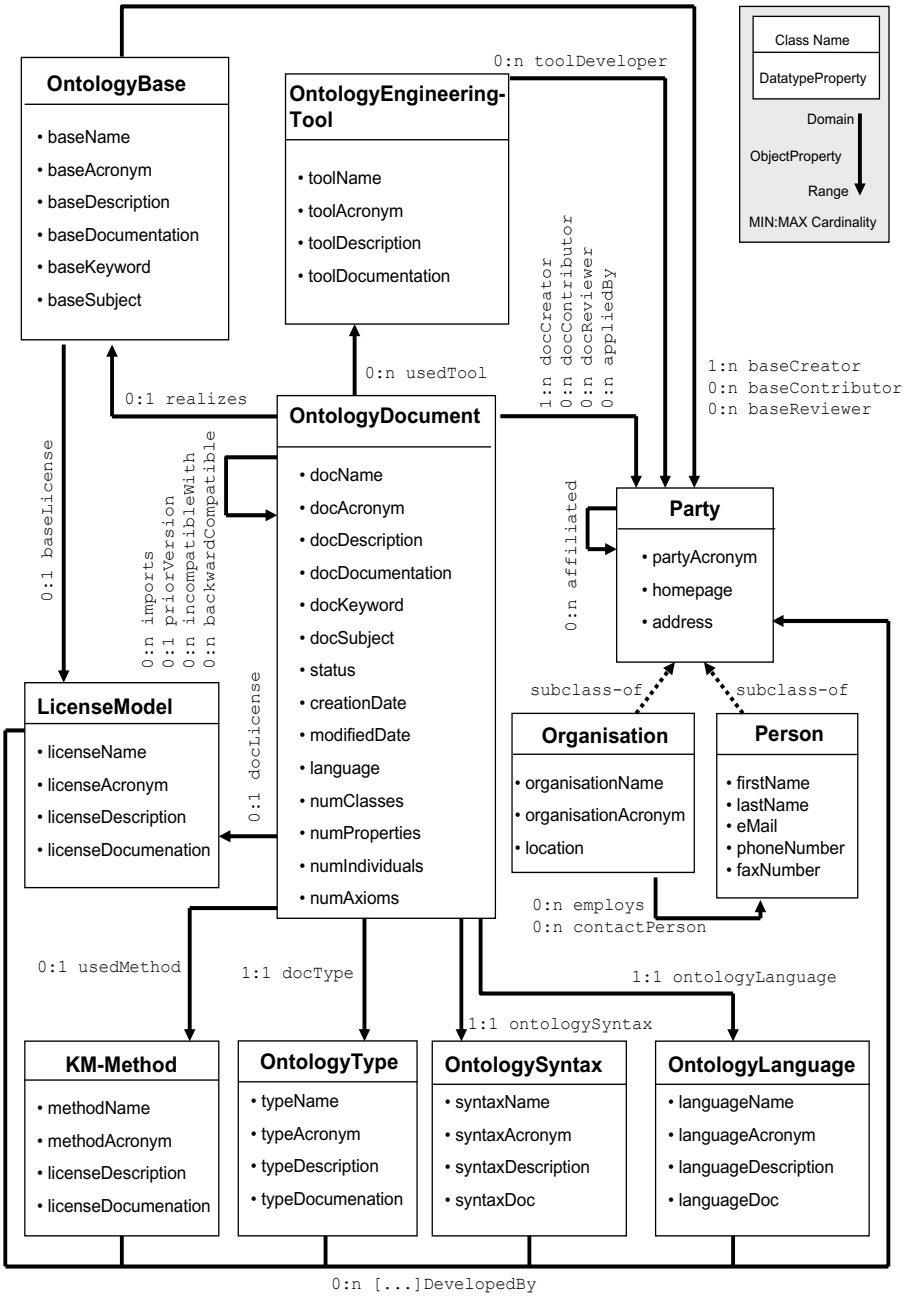


Fig. 2. General OMV overview

which can be realized by the class `LicenseModel`. So, that each `OntologyBase` or `OntologyDocument` is related to a pre-defined `LicenseModel`.

The presented OMV tries to model as much information about ontologies and the important aspects for ontology reuse as possible and at the same time intends to stay as simple as possible.

3 Applications for OMV

We now present running applications based on the proposed OMV. In detail, we present two complementary applications, namely the *decentralised* P2P system Oyster and the *centralised* metadata portal Onthology. Both applications have in common that they support single users and communities of users in *indentifying*, *reusing* and *providing* ontology metadata. As a consequence, they support the core idea of the Semantic Web and help to increase the applicability of ontologies.

In general, the two tools differ in their usage perspective and are appropriate for different tasks. However, as we will see, only the combined application of both tools will offer users the full potential of ontology metadata management.

3.1 Oyster – A Peer-to-Peer System for Sharing Ontologies

Overview. Oyster[4] is a Java-based system that exploits semantic web techniques in order to provide an innovative and useful solution for exchanging and reusing ontologies. For this purpose, Oyster provides facilities for managing, searching and sharing ontology metadata in a P2P network, thereby implementing the OMV proposal for the standard set of ontology metadata.

Oyster offers a user driven approach where each peer has its own local repository of ontology metadata and also has access to the information of others repositories, thus creating a virtual decentralized Ontology repository. The Oyster client on its own (e.g. disconnected from the P2P network) will already provide added value to its users as it will give developers an overview and search facilities of his/her own ontology metadata stored in its local repository. The goal is a decentralized knowledge sharing environment using Semantic Web technologies that allows developers to easily share ontology documents.

Functionalities. The Oyster system has been implemented as an instance of the Swapster system architecture [5]. It uses ontologies extensively in order to provide some of its main functions: importing data, formulating queries and processing answers.

Creating and importing metadata: Oyster enables users to create metadata about ontologies manually, as well as to import ontology files and to automatically extract the ontology metadata available, letting the user to fill in missing values. The ontology metadata entries are aligned and formally represented according to two ontologies: (1) the OMV ontology[6], (2) a topic hierarchy (i.e. the DMOZ topic hierarchy), which describes specific categories of subjects to define the domain of the ontology.

[4] `http://oyster.ontoware.org/`
[5] `http://swap.semanticweb.org/`
[6] `http://omv.ontoware.org/2005/05/ontology`

Formulating queries: A user can search for ontologies using simple keyword searches, or using more advanced, semantic searches. Here, queries are formulated in terms of these two ontologies. This means queries can refer to fields like name, acronym, ontology language, etc. or queries may refer to specific topic terms.

Routing queries: A user may query a single specific peer (e.g. their own computer, because they can have many ontologies stored locally and finding the right one for a specific task can be time consuming, or users may want to query another peer in particular because this peer is a known big provider of information), or a specific set of peers (e.g. all the member of a specific organization), or the entire network of peers (e.g. when the user has no idea where to search), in which case queries are routed automatically in the network.

Processing results: Finally, results matching a query are presented in a result list. The answer of a query might be very large, and contain many duplicates due to the distributed nature and potentially large size of the P2P network. Such duplicates might not be exactly copies because the semi structured nature of the metadata, so the ontologies are used again to measure the semantic similarity between different answers and to remove apparent duplicates. As proposed by the ontology metadata standard, all the different realizations of an ontology (ontology documents) can be grouped by the same ontology base to give a more organized view of the results.

3.2 Onthology – A Central Ontology Metadata Portal

As the importance of metadata increases with the number of existing ontologies, the storage and access to it becomes important as well. There exist mainly two kinds of storage facilities for ontology metadata. We present the conceptual design of a centralised ontology metadata portal and its implementation, so-called *Onthology* meaning an anthology of ontologies[7].

Actors. A main goal of a centralised metadata portal is to act as large evidence storage of metadata resp. their related ontologies to assure access, reuse and sharing, in the sense of the Semantic Web. We identified several different user roles for Onthology: The **visitor** is an anonymous user, he is allowed to browse the public content of the portal. A Visitor can become a **user** by completing an application form on the website. In order to avoid unnecessary administrative work, a User is added automatically to the membership database. Users can customize their portal, e.g. the content of their startpage or their bookmarks. If a user wants to add metadata to the portal, this submission has to be reviewed before it is published. Onthology works with a **review process** in order to ensure the quality of the metadata. **Reviewers** check the new submissions before it is published. The **technical administrator** is responsible for any other task mainly the maintenance of the portal.

Functionalities. Functionalities of Onthology can be separated into two groups based on the usage. Indeed, *basic functionalities* are provided to every user who accesses the repository and *sophisticated functionalities* for reviewers and administrators. The main operations a user can perform on the repository are the following.

[7] `http://www.onthology.org/`

- **Search:** *Query* and *browse* the repository
- **Submit:** *Provide* new metadata
- **Export:** *Download* (parts of) the repository.

The search and export can be performed by any visitor without being registered to the repository. Since providing new metadata is based on a certain community confidence, a visitor has to register at the repository to be become a registered user.

Architecture. A metadata portal mainly consists of *a large data repository* in which metadata can be stored. Exemplary, Sesame[8] or KAON[9] can be used as back-end metadata repository. Furthermore, *access* and in particular the *managament* of the repository must be guaranteed, too. Therefore, Onthology is based on SEAL, the AIFB conceptual architecture for building SEmantic portALs. In SEAL ontologies are key elements for managing community web sites and web portals. They support queries to multiple sources, but beyond that also intensive use of the schema information itself to allow for automatic generation of navigational views such as navigation hierarchies that appear as `has-part`-trees or `has-subtopic` trees in the ontology. In addition to that mixed ontology and content-based presentation is supported. Further information can be found at [7].

3.3 Discussion

Both presented applications are covering a variety of different tasks. Indeed, users who wants to store metadata individually similar to managing his personal favorite song list, a repository is required to which a user has full access and can perfom any operation (e.g. create, edit or delete metadata) without any consequences to other users. Exemplary, users from academia or industry might use a personal repository for a task-dependant investigation or ontology engineers, might use it during their ontology development process to capture information about different ontology versions. We argue, that a decentralised system is the technique of choice, since it allows the maximum of individuality while it still ensures exchange with other users.

Centralised systems allow to reflect long-term community processes in which some ontologies become well accepted for a domain or community and others become less important. Such well accepted ontologies and in particular their metadata need to be stored in a central metadata portal which can be accessed easily by a large number of users whereby the management procedures are well defined. Obviously, personal repositories are quite limited from this perspective.

Actually, the Oyster system and Onthology are not necessarily two completely separated repositories. Indeed, they are interconnected and they exchange metadata between each other. We are currently supporting the access of metadata stored in Onthology from any Oyster peer. However, accessing metadata in Onthology stored on Oyster peers is future work and requires more conceptual work, because the stored metadata within Onthology are based on a certain level of confidence among a community.

[8] `http://www.openrdf.org/`
[9] `http://kaon.semanticweb.org/`

The benefit of connecting both systems lies mainly in the simple use of existing ontology metadata information within Oyster. So, while users are applying or even developing their own ontologies they can manage their own metadata along with other existing metadata in one application (in Oyster). If some metadata entries from Oyster have reached a certain confidence, an import into Onthology can be performed easily. In combination, both systems ensure efficient and effective ontology metadata management for various use cases.

4 Related Work and Conclusion

We will briefly mention related metadata standards, including in particular those ones relevant for the Semantic Web. The **Dublin Core (DC)** metadata standard [2] is a simple yet effective element set for describing a wide range of networked resources. It includes two levels: Simple and Qualified. Simple DC comprises fifteen elements; Qualified DC includes an additional element as well as a group of element refinements (or qualifiers) that refine the semantics of the elements in ways that may be useful in resource discovery. **FOAF** [5], or "Friend Of A Friend", provides a way to create machine-readable Web homepages for people (their interests, relationships and activities), groups, companies and other kinds of things. To achieve this, FOAF project use the "FOAF vocabulary" to provide a collection of basic terms that can be used in these Web pages. The initial focus of FOAF has been on the description of people. The Semantic Web search engine **SWOOGLE** [3] makes use of particularly those metadata which can be extracted automatically. Our approach includes and extends this metadata vocabulary. Ideally, future versions of SWOOGLE would also take into account the additional vocabulary defined in OMV. There exist some similar approaches to our proposed solution to share ontologies, but in general they are limited in scope. E.g. the **DAML ontology library** [9] provides a catalog of DAML ontologies that can be browsed by different properties. The **FIPA ontology service** [11] defines an agent wrapper of open knowledge base connectivity. Finally we mention the **SchemaWeb Directory** [4] that is a repository for RDF schemas expressed in RDFS, OWL and DAML+OIL. While the goal of SchemaWeb is similar to that of Onthology, its metadata ontology is less comprehensive.

The term *ontology base* is used in different context in DOGMA[8]: A DOGMA ontology consists of an ontology base that holds sets of intuitive context-specific conceptual relations and a layer of "relatively generic" ontological commitments that hold the domain rules. In contrast, in our work we use the term *ontology base* to represent the conceptualisation of an ontology.

To conclude, reusing existing ontologies is a key issue for sharing knowledge on the Semantic Web. Our contribution aims at facilitating reuse of ontologies which are previously unknown for ontology developers by providing an Ontology Metadata Vocabulary (OMV) and two prototypical applications for decentralized (Oyster) and centralized (Onthology) sharing of ontology metadata based on OMV.

OMV has been proposed and discussed in the industry area of the EU thematic network of excellence Knowledge Web (KWeb). Next steps include the standardization of OMV in a wider scope by particularly including non-KWeb parties in this process,

followed by a close cooperation with tool providers for ontology engineering environments and applications providers for e.g. ontology based search engines to enhance their tools with support for OMV. The agreement and application of a standard on a global level will greatly facilitate the reuse of ontologies for all participating parties.

Acknowledgments. Research reported in this paper has been partially financed by EU in the IST project Knowledge Web (FP6-507482).

References

1. D. Brickley and R. V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Rec. 10 February 2004, 2004. available at `http://www.w3.org/TR/rdf-schema/`.
2. Dublin Core. `http://dublincore.org/`.
3. Li Ding et al. Swoogle: A search and metadata engine for the semantic web. In *In Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management*, pages 58–61, November 2004.
4. SchemaWeb directory. `http://www.schemaweb.info/`.
5. FOAF. `http://www.foaf-project.org/`.
6. J. Hartmann and R. Palma. OMV - Ontology Metadata Vocabulary for the Semantic Web, 2005. v. 1.0, available at `http://omv.ontoware.org/`.
7. J. Hartmann and Y. Sure. An infrastructure for scalable, reliable semantic portals. *IEEE Intelligent Systems*, 19(3):58–65, May/June 2004.
8. Spyns Peter, Meersman Robert, and Jarrar Mustafa. Data modelling versus ontological engineering. In *SIGMOD Record Special Issue on Semantic Web, Database Management and Information Systems*, volume 31, pages 7–12, December 2002.
9. DAML Ontology Repository. `http://www.daml.org/ontologies/`.
10. M. K. Smith, C. Welty, and D. McGuinness. OWL Web Ontology Language Guide, 2004. W3C Rec. 10 February 2004, available at `http://www.w3.org/TR/owl-guide/`.
11. H. Suguri et al. Implementation of fipa ontology service. In *Proc. of the Workshop on Ontologies in Agent Systems, 5th Int. Conf. on Autonomous Agents Montreal, Canada*, 2001.
12. C. Tempich, H. S. Pinto, Y. Sure, and S. Staab. An argumentation ontology for distributed, loosely-controlled and evolving engineering processes of ontologies (diligent). In A. Gez-Pez and J. Euzenat, editors, *2nd European Semantic Web Conference, ESWC 2005*, volume 3532 of *LNCS*, pages 241–256, Heraklion, Crete, Greece, MAY 2005. Springer.

Ontological Foundation for Protein Data Models

Amandeep S. Sidhu[1], Tharam S. Dillon[1], and Elizabeth Chang[2]

[1] Faculty of Information Technology, University of Technology, Sydney, Australia
{asidhu, tharam}@it.uts.edu.au
[2] School of Information Systems, Curtin University of Technical University, Perth, Australia
Elizabeth.Chang@cbs.curtin.edu.au

Abstract. In this paper we proposed a Protein Ontology to integrate protein data and information from various Protein Data Sources. Protein Ontology provides the technical and scientific infrastructure and knowledge to allow description and analysis of relationships between various proteins. Protein Ontology uses relevant protein data sources of information like PDB, SCOP, and OMIM. Protein Ontology describes: Protein Sequence and Structure Information, Protein Folding Process, Cellular Functions of Proteins, Molecular Bindings internal and external to Proteins, and Constraints affecting the Final Protein Conformation. We also created a database of 10 Major Prion Proteins available in various Protein data sources, based on the vocabulary provided by Protein Ontology. Details about Protein Ontology are available online at http://www.proteinontology.info/.

1 Introduction

A large variety of proteins have been deduced from the various genome projects and within them have been identified conserved or variant regions, functional and structural elements, features and domains. The continuum of life forms has become clearer and differences between species measurable. Novel biocatalysts and parameters relating structure to function have been identified from diversity of living organisms, and the network of molecular interactions and complex biological processes has become available for modelling. The primary tools for drug discovery are now the classification of chemical compounds, proteins and targets into functional groups, identification of relations between distant chemical targets & cells, drug effects on cells and lastly the knowledge of chemical structures and properties. Advanced protein engineering with the help of computer science has proven a sophisticated aid in the development of new biocatalysts, therapeutics and diagnostic tools. Advanced methods like high-throughput crystallography and nuclear magnetic resonance (NMR) have accelerated the resolution of new protein structures, and the modelling of macromolecules have been improved new groups of 3D protein structures.

These developments increased the importance of proteomics in heath care. Protein Informatics is a multidisciplinary field that is a synergy of proteomics process, bioinformatics and computational biology. The main objective of Protein Informatics

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 916 – 925, 2005.
© Springer-Verlag Berlin Heidelberg 2005

is to provide a framework for developing, integrating and sharing knowledge present in various protein data sources. The advances in information and communication technologies coupled with increased knowledge about genes and proteins have opened new perspectives for study of protein complexes. There is a growing need to integrate the knowledge about various protein complexes for effective disease prevention mechanisms, individualized medicines and treatments and other accepts of healthcare. The exploitation of data from bioinformatics, medical informatics, medical imaging and clinical data requires a new and synergetic approach that enables a bilateral dialogue between these scientific disciplines, and integration in terms of data, methods, technology, tools and applications.

The proposed Protein Ontology provides the technical and scientific infrastructure and knowledge to allow evidence based description and analysis of relationships between proteins and other macromolecules. Protein Ontology uses all relevant protein data sources of information. The sources include new proteome information resources like PDB [1, 2, 3, 4] and SCOP [5, 6], as well as classical sources of information where information is maintained in a knowledge base of scientific text files like OMIM [7] and from various published scientific literature in various journals. PDB [1, 2, 3, 4] mainly provides protein entry and structure information for the Protein Ontology. SCOP [5, 6] provides the structural domain classification system that is widely used in proteomics. OMIM [7] is a knowledge base that provides texts and literature on various gene defects affecting the genes that make proteins. The information about the newly proposed functional domain classification system is gathered from various scientific literature and texts. On the whole Protein Ontology is an effort to seamlessly integrate all the data and knowledge about Proteins, to provide a data specification for new data representation and existing mining of existing data.

2 Ontology Foundations

Traditionally the knowledge base in biology has resided within the heads of experienced biologists and scientists who devoted study and time to become experts in their particular domain. This approach worked well in past when considerable effort was needed to tease data out of biological experiments, the flow of data was not so great to overwhelm the expert. However this situation is rapidly changing, many protein complexes are appearing each year and new experimental techniques are providing information on protein interactions. Not only is the rate of data acquisition growing exponentially but also a single experiment can collect data on huge range of molecules that would need many domain experts to interpret. There is therefore a need to create systems that can apply knowledge of domain experts to biological data. It is not envisaged that such systems will perform better than human experts; however they could play a crucial role in filtering the flood of data to point where human experts could again apply their knowledge. This then raises the questions, in particular how concepts and their relationships can be captured in ways that make them computationally available and traceable.

Ontology is a system that describes concepts and the relationships between them. Therefore, we proposed to build ontology for the Proteomics Domain. It is important to point out that this is an integration of data formats for representation. The Protein Ontology is an ontology based integration of heterogenous protein and biological data sources. Protein Ontology converts the enormous amounts of data collected by geneticists and molecular biologists into information that scientists, physicians and other health care professionals and researchers can use to easily understand the mapping of relationships inside protein molecules, interaction between two protein molecules and interactions between protein and other macromolecules at cellular level. Protein Ontology also helps to codify proteomics data for analysis by researchers.

A considerable body of research in the area of knowledge representation has shown that ontology must necessarily reflect a specific view of the data. Traditionally, ontologies have been represented using static models [8]. These can assist in exchanging knowledge at a purely terminological or syntactic level, but can suffer due to difficulties of interpretation; the relationships in the model rely solely on the prospective of the modeller. If we are to share knowledge, a clearer semantics is required. Full interaction with ontology requires, in addition a notion of functionality or reasoning the ontology can provide. Frame representations provide a precise, definitional framework to capture concepts and relationships between them. Frame formalism has been used to model biological data in EcoCyc Encyclopedia of E.Coli genes and metabolism [9]. The representation is however, static in the sense that kind-of hierarchy is asserted by the modeller, rather than deduced by the system from the descriptions of the concepts.

Description Logics (DL) [10] is an example of knowledge representation language. DL provides a language for capturing declarative knowledge about a domain and a classifier that allows reasoning about that knowledge. Information captured using DL is classified in a rich hierarchy of concepts and their inter-relationships. DL is compositional and dynamic, relying on notions of classification, reasoning, consistency, retrieval and querying. In this paper we investigated use of OWL Description Logics (OWL-DL) for making Protein Ontology using Protégé OWL Plug-in. The OWL-DL is chosen for its following features:

1. OWL-DL is flexible and powerful enough to capture and classify biological concepts of proteins in a consistent and principled fashion.
2. OWL-DL is used to construct protein ontology that can be used for making inferences from proteomics data.

3 Related Work

In this section we will discuss various biomedical ontology works related to Protein Ontology. Gene Ontology (GO) [11] defines a hierarchy of terms related to genome annotation. GO is a structured network consisting of defined terms and relationships that describe Molecular Functions, Biological Processes, and Cellular Components of Genes. GO is clearly defined and modelled for numerous other biological ontology

projects [12]. So far GO has been used to describe the genes of several model organisms *(Saccharomyces cerevisiae, Drosophila melanogaster, Mus musculus and others)*.

RiboWEB [13] is an online data resource for Ribosome, a vital cellular apparatus. It contains a large knowledge base of relevant published data and computational modules that can process this data to test hypotheses about ribosome's structure. The system is built around the concept of ontology. Diverse types of data taken principally from published journal articles is represented using a set of templates in the knowledge base, and the data is linked to each other with numerous connections.

Protein Data Bank (PDB) has recently released versions of the PDB Exchange Dictionary and the PDB archival files in XML format collectively named PDBML [14]. The representation of PDB data in XML builds from content of PDB Exchange Dictionary, both for assignment of data item names and defining data organization. PDB Exchange and XML Representations use same logical data organization. A side effect of maintaining a logical correspondence with PDB Exchange representation is that PDBML lack the hierarchical structure characteristic of XML data.

PRONTO [15] is a directed acyclic graph (DAG) based ontology induction tool that constructs a protein ontology including protein names found in MEDLINE abstracts and in UNIPROT. It is a typical example typical example of mining literature and data sources. It can't be classified as protein ontology as it only represents relationship between protein literatures and does not formalize knowledge about protein synthesis process. Ontology for Protein Domain must contain terms or concepts relevant to protein synthesis, describing Protein Sequence, Structure and Function and relationships between them. While defining PO we made an effort to emulate the protein synthesis and describe concepts and relationships describing it.
There is a need for more agreed-upon semantical standard to describe protein data. PO addresses this issue by providing clear and unambiguous definitions of all major biological concepts of protein synthesis process and relationship between them using OWL. The use OWL in PO provides a unified controlled vocabulary both for annotation data types and for annotation data.

4 Protein Ontology Overview

We defined a Protein Ontology (PO) [16, 17, 18, 19, 20] that provides a common structured vocabulary for researchers who need to share knowledge in proteomics domain. PO consists of concepts (or classes), which are data descriptors for proteomics data and the relations among these concepts. PO has (1) a hierarchical classification of concepts represented as classes, from general to specific; (2) a list of attributes related to each concept, for each class; and (3) a set of relations between classes to link concepts in ontology in more complicated ways then implied by the hierarchy, to promote reuse of concepts in the ontology. At the moment PO currently contains 92 *concepts* or classes and 261 *attributes* or properties. The structure of PO provides the concepts necessary to describe individual proteins, but does not contain individual protein themselves. The underlying XML database based on PO acts as

instance store for the PO. The Database consists of 17550 instances of 10 major prion proteins for various concepts defined in PO. PO provides a structured vocabulary description for protein domains that can be used to describe cellular products in any organism. Protein Ontology Framework describes: (1) Protein Sequence and Structure Information, (2) Protein Folding Process, (3) Cellular Functions of Proteins, (4) Molecular Bindings internal and external to Proteins and (5) Constraints affecting the Final Protein Conformation.

The Main Class of Protein Ontology is ProteinOntology. For each Protein that is entered into the knowledge base of protein ontology, submission information is entered into ProteinOntology Class. ProteinOntologyID has format like "PO000000005". There are seven subclasses of ProteinOntology (PO), called Generic Classes that are used to define complex concepts in other PO Classes: Residues, Chains, Atoms, Family, AtomicBind, Bind, and SiteGroup. Concepts from these generic classes are reused in various other PO Classes for definition of Class Specific Concepts. Details and Properties of Residues in a Protein Sequence are defined by instances of Residues Class. Instances of Chains of Residues are defined in Chains Class. All the Three Dimensional Structure Data of Protein Atoms is represented as instances of Atoms Class. Defining Chains, Residues and Atoms as individual classes has the benefit that any special properties or changes affecting a particular chain, residue and atom can be easily added. Protein Family class represents Protein Superfamily and Family Details of Proteins. Data about binding atoms in Chemical Bonds like Hydrogen Bond, Residue Links, and Salt Bridges is entered into ontology as an instance of AtomicBind Class. Similarly the data about binding residues in Chemical Bonds like Disulphide Bonds and CIS Peptides is entered into ontology as an instance of Bind Class. All data related to site groups of the active binding sites of Proteins is defined as instances of SiteGroup Class. Representation of Instances of Residues and Chains of Residues are shown as follows:

```
<Residues>
        <Residue>LEU</Residue>
        <ResidueName>LEUCINE</ResidueName>
        <ResidueProperty>1-LETTER CODE: L; FORMULA: C6 H13 N1 O2;
    MOLECULAR WEIGHT: 131.17</ResidueProperty>
</Residues>

<Chains>
        <Chain>D</Chain>
        <ChainName>CHAIN D</ChainName>
</Chains>
```

The Root Class for definition of Protein Complexes in the Protein Ontology is ProteinComplex. The Protein Complex Definition defines one or more Proteins in the Complex Molecule. There are six main subclasses within ProteinComplex class: Entry, Structure, StructuralDomains, FunctionalDomains, ChemicalBonds, and Constraints. These classes define sequence, structure, function, and chemical bindings present in the Protein Complex. The Complete Class Hierarchy of Protein Ontology (PO) is shown in Figure 1**.** More detailed UML Diagrams for PO are available at the website.

- **ProteinOntology**
  - **AtomicBind**
  - **Atoms**
  - **Bind**
  - **Chains**
  - **Family**
  - **ProteinComplex**
    - **ChemicalBonds**
      - **CISPeptide**
      - **DisulphideBond**
      - **HydrogenBond**
      - **ResidueLink**
      - **SaltBridge**
    - **Constraints**
      - **GeneticDefects**
      - **Hydrophobicity**
      - **ModifiedResidue**
    - **Entry**
      - **Description**
      - **Molecule**
      - **Reference**
    - **FunctionalDomains**
      - **ActiveBindingSites**
      - **BiologicalFunction**
        - **PathologicalFunctions**
        - **PhysiologicalFunctions**
      - **SourceCell**
    - **StructuralDomains**
      - **Helices**
        - **Helix**
          - **HelixStructure**
      - **OtherFolds**
        - **Turn**
          - **TurnStructure**
      - **Sheets**
        - **Sheet**
          - **Strands**
      - **Structure**
        - **ATOMSequence**
        - **UnitCell**
  - **Residues**
  - **SiteGroup**

Fig. 1. Class Hierarchy of Protein Ontology

5 Protein Ontology Implementation

OWL-DL relies on notions classification, reasoning, and consistency. These notions are applied in the making of Protein Ontology by defining new concepts or classes from defined generic concepts or classes. The concepts derived from generic concepts are placed precisely into class hierarchy of Protein Ontology to completely represent information defining a protein. As the OWL representation used in Protein Ontology is an XML-Abbrev based (Abbreviated XML Notation), it can be easily transformed to the corresponding RDF and XML formats without much effort using the available converters.

To understand the reuse of concepts in Protein Ontology, here are some of the examples. ATOMSequence instance is constructed using generic concepts of Chains, Residues, and Atoms. The reasoning is already there in the underlying relationships and hierarchy of Protein Data, as each Chain in a Protein represents a sequence of Residues, and each Residue is defined by a number of three dimensional atoms in the Protein Structure. The Structure Class of Protein Ontology that is used to define ATOMSequence, with references to definitions of Chain and Residues.

```
<ATOMSequence>
        <ProteinOntologyID>PO0000000004</ProteinOntologyID>
        <_ATOM_Chain>A</_ATOM_Chain>
        <_ATOM_Residue>ARG</_ATOM_Residue>
        <AtomID>364</AtomID>
        <Atom>HE</Atom>
        <ATOMResSeqNum>148</ATOMResSeqNum>
        <X>-23.549</X>
        <Y>3.766</Y>
        <Z>-0.325</Z>
        <Occupancy>1</Occupancy>
        <TempratureFactor>0</TempratureFactor>
        <Element>H</Element>
</ATOMSequence>
```

Similarly Secondary Structure elements of Protein Structure like helices, sheets, and short loops can also be represented using generic concepts of Chains and Residues. The hierarchy used in a Helices Instance of Protein Ontology differentiates general information about the Helices and the Helix Structure comprising of Chains and Residue Sequences.

```
<Helices>
        <ProteinOntologyID>PO0000000002</ProteinOntologyID>
        <_StrDomain_SuperFamily>HAMSTER</_StrDomain_SuperFamily>
        <_StrDomain_Family>PRION PROTEINS</_StrDomain_Family>
        <HelixID>1</HelixID>
        <HelixNumber>1</HelixNumber>
        <HelixClass>Right Handed Alpha</HelixClass>
        <HelixLength>10</HelixLength>
        <HelixStructure>
```

```
            <_Helix_Chain>A</_Helix_Chain>
            <_Helix_InitialResidue>ASP</_Helix_InitialResidue>
            <HelixInitialResidueSeqNum>144</HelixInitialResidueSeqNum>
            <_Helix_EndResidue>ASN</_Helix_EndResidue>
            <HelixEndResidueSeqNum>153</HelixEndResidueSeqNum>
        </HelixStructure>
    </Helices>
```

Other secondary structures like sheets and loops are represented using concepts of chains and residues in the similar way. The Sheet Structures in Proteins are composed of various Strands and is represented as follows using Protein Ontology.

```
    <Sheets>
            <ProteinOntologyID>PO0000000001</ProteinOntologyID>
            <_StrDomain_SuperFamily>MOUSE</_StrDomain_SuperFamily>
            <_StrDomain_Family>PRION PROTEINS</_StrDomain_Family>
            <SheetID>S1</SheetID>
            <NumberStrands>2</NumberStrands>
            <Strands>
                <StrandNumber>2</StrandNumber>
                <_Strand_Chain>NULL</_Strand_Chain>
                <_Strand_IntialResidue>VAL</_Strand_IntialResidue>
                <StrandIntialResidueSeqNum>161</StrandIntialResidueSeqNum>
                <_Strand_EndResidue>ARG</_Strand_EndResidue>
                <StrandEndResidueSeqNum>164</StrandEndResidueSeqNum>
                <StrandSense>ANTI-PARALLEL</StrandSense>
    </Strands></Sheets>
```

Again the various chemical bonds used to bind various substructures in a complex protein structure are defined using generic concepts of Bind and Atomic Bind. The Chemical Bonds that have Binding Residues reuse the generic concept of Bind. In defining the generic concept of Bind in Protein Ontology we again reuse the generic concepts of Chains and Residues. Similarly the Chemical Bonds that have Binding Atoms reuse the generic concept of AtomicBind. In defining the generic concept of AtomicBind we reuse the generic concepts of Chains, Residues and Atoms.

```
    <CISPeptides>
            <ProteinOntologyID>PO0000000003</ProteinOntologyID>
            <_Bind_Chain_1>H</_Bind_Chain_1>
            <_Bind_Residue_1>GLU</_Bind_Residue_1>
            <BindResSeqNum_1>145</BindResSeqNum_1>
            <_Bind_Chain_2>H</_Bind_Chain_2>
            <_Bind_Residue_2>PRO</_Bind_Residue_2>
            <BindResSeqNum_2>146</BindResSeqNum_2>
            <AngleMeasure>-6.61</AngleMeasure>
            <Model>0</Model>
    </CISPeptides>
```

A XML Database of 10 Major Prion Proteins available in various Protein data sources, based on the vocabulary provided by Protein Ontology is available on the PO

website. Soon we will have all the 57 Prion Proteins known to exist, and user interfaces to browse and query the database. The XML database currently contains 24 tables, 261 attributes and 17550 instances. Prion Protein is a membrane bound protein of 253 amino acid residues in length that is normally found in neurons and several other cell types. The abnormal Prion Protein is resistant to digestion with enzymes that breaks down normal proteins, and accumulates in the brain. Abnormal Prion Proteins are the major cause of various Human Prion Diseases in Brain like Fatal Familial Insomnia. Recently, discovery of Interesting Properties of Prion Proteins encouraged Scientists to understand Prion Proteins for finding cure to various Human Brain Diseases. Building a XML Data Source based on PO will assist in discovery process.

6 Conclusion

Protein Ontology (PO) provides a unified vocabulary for capturing declarative knowledge about protein domain and to classify that knowledge to allow reasoning. Information captured by PO is classified in a rich hierarchy of concepts and their inter-relationships. PO is compositional and dynamic, relying on notions of classification, reasoning, consistency, retrieval and querying. In PO the notions classification, reasoning, and consistency are applied by defining new concepts or classes from defined generic concepts or classes. The concepts derived from generic concepts are placed precisely into class hierarchy of Protein Ontology to completely represent information defining a protein complex. As the OWL representation used in Protein Ontology is an XML-Abbrev based (Abbreviated XML Notation), it can be easily transformed to the corresponding RDF and XML formats without much effort using the available converters. Our Protein Ontology (PO) is the first ever work to integrate protein data based on data semantics describing various phases of protein structure. PO helps to understand structure, cellular function and the constraints that affect protein in a cellular environment. The attribute values in the PO are not defined as text strings or as set of keywords. Most of the Values are entered as instances of Concepts defined in Generic Classes. We defined a XML Database of Human Prion Proteins based on PO, gathering information about various Prion Proteins from various data sources. For Protein Functional Classification, in addition to presence of domains, motifs or functional residues, following factors are relevant: (a) similarity of three dimensional protein structures, (b) proximity to genes (may indicate that proteins they produce are involved in same pathway), (c) metabolic functions of organisms and (d) evolutionary history of the protein. At the moment PO's Functional Domain Classification does not address the issues of proximity of genes and evolutionary history of proteins. These factors will be added in future to complete the Functional Domain Classification System in PO. The Constraints defined in PO are not mapped back to protein sequence, structure and function they affect. Achieving this in future will inter-link all the concepts of PO.

References

[1] Weissiga, H. And P. E. Bourne (2002). "Protein structure resources." Biological Crystallography D58: 908-915.

[2] Westbrook, J., Z. Feng, et al. (2002). "The Protein Data Bank: unifying the archive." Nucleic Acid Research 30(1): 245-248.

[3] Bhat, T. N., P. E. Bourne, et al. (2001). "The PDB data uniformity project." Nucleic Acid Research 29(1): 214-218.

[4] Bernstein, F. C., T. F. Koetzle, et al. (1977). "The Protein Data Bank: a computer-based archival file for macromolecular structures." Journal of Molecular Biology 112(3): 535-42.

[5] Conte, L. L., B. Ailey, et al. (2000). "SCOP: a Structural Classification of Proteins database." Nucleic Acids Research 28(1): 257-259.

[6] Murzin, A. G., S. E. Brenner, et al. (1995). "SCOP: A Structural Classification of Proteins Database for the Investigation of Sequences and Structures." Journal of Molecular Biology 247: 536–540.

[7] mckusick, V. A. (2000). "Online Mendelian Inheritance in Man", OMIM. Baltimore, MD, Johns Hopkins University, National Center for Biotechnology Information, and National Library of Medicine.

[8] Schulze-Kremer, S. (1998). "Ontologies for Molecular Biology." Proceedings of Third Pacific Symposium on Biocomputing, Hawaii, AAAI Press: 693-704.

[9] Karp, P., M. Riley, et al. (1998). "Ecocyc: Electronic Encyclopedia of E.coli genes and metabolism." Nucleic Acids Research 26, 50.

[10] Borgida, A. (1995) "Description Logics in Data Management." IEEE Transactions on Knowledge and Data Engineering, 7, 671-682.

[11] GO. (2001). "Creating the Gene Ontology Resource: Design and Implementation." Genome Research 11: 1425-1433.

[12] GO. (2004). "Gene Ontology: looking backwards and forwards." Genome Biology 6(1): 103.1-103.4.

[13] Altmann, R. B., M. Bada, et al. (1999). "riboweb: An Ontology-Based System for Collaborative Molecular Biology." IEEE Intelligent Systems (SEPTEMBER/OCTOBER 1999): 68-76.

[14] Westbrook, J., N. Ito, et al. (2005). "PDBML: The Representation of Archival Macromolecular Structure Data in XML." Bioinformatics 21(7): 988-992.

[15] Mani, I., Z. Hu, et al. (2004). PRONTO: A Large-scale Machine-induced Protein Ontology. 2nd Standards and Ontologies for Functional Genomics Conference (SOFG 2004), UK.

[16] Sidhu, A. S., T. S. Dillon, et al. (2005). An Ontology for Protein Data Models. 27th annual international conference of the ieee engineering in medicine and biology society 2005 (IEEE EMBC 2005). Shanghai, China. IEEE Press.

[17] Sidhu, A. S., T. S. Dillon, et al. (2005). Ontology-based Knowledge Representation of Protein Data. 3rd International IEEE Conference on Industrial Informatics, Perth, Australia, IEEE CS Press.

[18] Sidhu, A. S., T. S. Dillon, et al. (2005). Protein Ontology: Vocabulary for Protein Data. 3rd IEEE International Conference on Information Technology and Applications. Sydney, Australia, IEEE CS Press.

[19] Sidhu, A. S., T. S. Dillon, et al. (2004). A Unified Representation of Protein Structure Databases (Book Section). Biotechnological Approaches for Sustainable Development. M. S. Reddy and S. Khanna. Mumbai, India, Allied Publishers Pvt. Ltd.: 396-408.

[20] Sidhu, A. S., T. S. Dillon, et al. (2004). An XML based semantic protein map. Data Mining 2004. A. Zanasi, N. F. F. Ebecken and C.A.Brebbia. Malaga, Spain, WIT Press, Southampton, UK. 10: 51-60.

SWQL – A Query Language for Data Integration Based on OWL

Patrick Lehti and Peter Fankhauser

Fraunhofer IPSI,
Dolivostr. 15, Darmstadt,
Germany
{Patrick.Lehti, Peter.Fankhauser}@ipsi.fraunhofer.de

Abstract. The Web Ontology Language OWL has been advocated as a suitable model for semantic data integration. Data integration requires expressive means to map between heterogeneous OWL schemas. This paper introduces SWQL (Semantic Web Query Language), a strictly typed query language for OWL, and shows how it can be used for mapping between heterogeneous schemas. In contrast to existing RDF query languages which focus on selection and navigation, SWQL also supports construction and user-defined functions to allow for instantiating integrated global schemas in OWL.

1 Introduction

Many applications especially from the Semantic Web community assume the existence of an ontology that models the domain of the application in an integrated way. However, the majority of existing data sources are not modelled in a way that relates instances directly to ontology classes and properties (like RDF can do), but are modelled as relations or as XML. These data models come with their own schema and query languages, like relational schema and SQL or XML schema and XQuery respectively. In order to access such sources via ontologies, we have developed the query language SWQL (Semantic Web Query Language). SWQL queries are formulated according to an OWL (Web Ontology Language) ontology, the underlying graph-based data model of SWQL can easily be mapped and implemented on top of the relational or XML data model. This allows to pose queries based on an ontology, independent of the underlying data model.

SWQL uses an XQuery-like syntax and is like XQuery a declarative, fully compositional, strictly typed, functional query language. It supports navigation and selection via its sublanguage SWQLPath (similar to XPath) and unlike most RDF query languages [1] it supports joins and construction. Together with user-defined functions, SWQL can be used as mapping language for data integration, where such mappings are implemented as SWQL function libraries. The strict typing of SWQL allows to check the consistency of such a mapping, i.e. given the input and output schema it can be garantueed that the mapping produces valid instances of the output schema.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 926–935, 2005.
© Springer-Verlag Berlin Heidelberg 2005

This paper presents the query language SWQL and shows its application in the field of data integration. Section 2 briefly introduces OWL and a definition of a data model for OWL instances. Section 3 presents the main features of the SWQL language. Section 4 shows its application as mapping language for data integration. Section 5 compares with related work and Section 6 concludes.

2 The Web Ontology Language OWL

OWL[2] is an emerging standard for a "Web Ontology Language" by the W3C. It is a successor of the ontology language DAML+OIL[3]. OWL allows to define classes, properties and literals (called datatypes). Properties define relationships between classes or between classes and datatypes. An example OWL schema for bibliographic data can be seen in Figure 1.

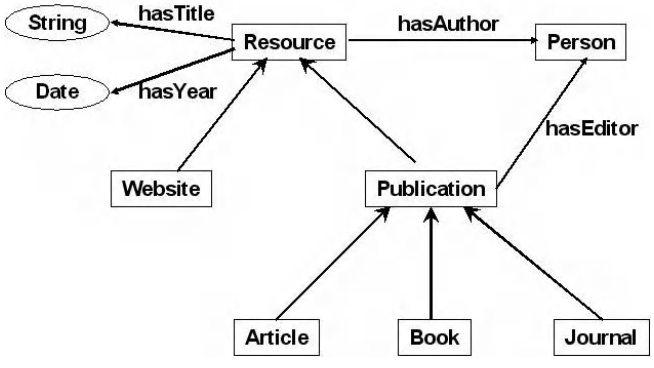


Fig. 1. OWL example

This schema contains a general class "Resource", which can be a "Website" or a "Publication". A "Publication" itself can be either an "Article", a "Book" or a "Journal". Every "Resource" can have a title, a year and authors, which are of class "Person". A "Publication" can also have a "Person" as editor. OWL schemas are graph structures, which can be serialized based on RDF[4].

2.1 A Data Model for OWL Instances

The graph structure of OWL schemas holds also for instances. This section formally defines the data model for OWL instances.

A graph consists of items. Items are either nodes or properties:

$$Item ::= Node | Property$$

A node is either an object or a literal:

$$Node ::= Object | Literal$$

The following functions are defined on items, the function declarations use an XQuery[5] syntax. Every item of a graph has an associated type from the OWL ontology. The "type" function returns this associated type as a qualified name as defined in XML Schema[6].

```
type($i as Item) as xs:QName
```

Every property connects two nodes. The "domain" of a property is always an object, the "range" can be any node.

```
domain($p as Property) as Object
```

```
range($p as Property) as Node
```

Every object has an unique id that is represented as string, this can be accessed via the "id" function.

```
id($o as Object) as xs:string
```

All objects *o* may have properties, which are edges outgoing from *o*. The "properties" function returns a set of such properties.

```
properties($o as Object) as Property*
```

Every literal has a value, which is an atomic type as defined in the XQuery 1.0 and XPath 2.0 specifications[5,7].

```
value($l as Literal) as xdt:anyAtomicType
```

This data model is very common and similar to the object-oriented data model, where the items are called Object, Field and Literal, to the ER model, where they are called Entity, Relationship and Literal and also to the RDF data model, where they are called Resource, Property and Literal. A mapping of this model to the relational model is similar as the mapping from the ER model. A mapping to the XML model is described in [8].

In order to use such a data model for a query language it is also necessary to define a notion of node sets. Node sets are the result of every SWQL expression. SWQL node sets contain nodes, where objects with the same id occur only once, whereas literals may occur more than once. SWQL node sets also preserve the order during construction. On node sets the function "get" returns the node at a specific position and the function "size" returns the number of nodes in the set.

```
get($n as Node*, $i as xs:int) as Node?
```

```
size($n as Node*) as xs:int
```

2.2 Integration of OWL Schemas

When two data sources with different schemas need to be integrated, the problem of schema integration arises, i.e. heterogenities between the schemas must be overcome by mappings. Such heterogenities can be:

- Different vocabulary: For equal or similar concepts different vocabulary is used, e.g. in one source a property is called "hasAuthor" the other source calls the equivalent property "writer".
- Different structure: The same information is represented at a different position in the ontology. E.g. the fname and lname properties in Figure 2 represent the same information but have an additional "Name" concept as domain.

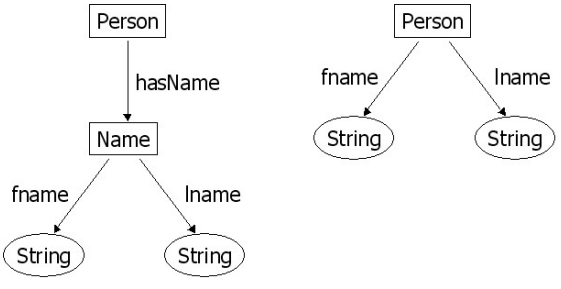


Fig. 2. Different Path in OWL

- Level of detail: the same information can be modelled at a different level of detail. See Figure 3, where the name of person is ones modelled as first name and last name and ones has a single string.

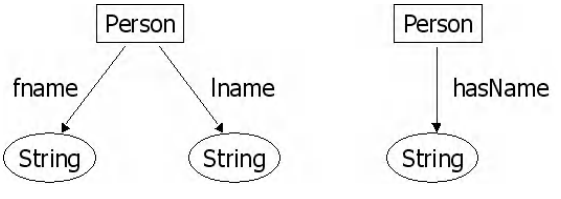


Fig. 3. Different Level of Detail in OWL

Whereas different vocabulary can be mapped by simple OWL built-in features like *equivalentClass* and *equivalentProperty*, different structures already need some mechanism to map paths. The "level of detail"-problem can only be solved with the help of powerful query languages, that allow to rearrange values and objects to build up the required structure of the target schema.

Beside of the heterogenities between the schemas, an integration system must also guarantee closed views on the local data sources, i.e., whenever a data source is accessed via a target schema and a corresponding mapping to the local schema, every item of the result graph must be part of the target schema. This is in particular a problem for graph data models, because navigating the result graph potentially involves navigating the complete source graph, which requires dynamically constructing target schema objects out of source objects.

3 The Query Language SWQL

SWQL uses OWL as its type system, a graph-based data model (as described in Section 2.1) and an XQuery-like syntax. A SWQL query consists of a prolog, which may contain declarations for importing modules (function libraries) and ontologies and for defining variables and user-defined functions and it consists of the query expression. SWQL expressions are basically SWQLPaths, FLWOR expressions or constructors. A full specification of the SWQL language can be found in [9].

3.1 SWQLPath

SWQLPath is a navigation and selection language on the SWQL data model similar to XPath [7] for the XQuery data model. This path language provides the possibility to address paths in a schema that can e.g. later be used for path mappings. A SWQLPath consists of a NodeExpr and zero or more following StepExpr separated by "/".

```
SWQLPath ::= NodeExpr ("/" StepExpr)*
NodeExpr ::= PrimaryExpr Predicates
StepExpr ::= PropertyTest Predicates
PropertyTest ::= QName | "*"
Predicates   ::= ("[" Expr "]")*
```

The NodeExpr selects an intial set of nodes from the graph, which are further filtered by predicates or navigated by StepExprs. NodeSets can be selected e.g. by variables, function calls or so-called node tests. The following NodeTest selects all Objects that are instances of the OWL class "Publication".

```
Publication()
```

Predicates filter the set of current nodes, e.g. the following query selects all Publications, whose publication year is "1998".

```
Publication()[hasYear = "1998"]
```

StepExprs navigate from the set of current nodes via property type names and return a new set of nodes. These are called PropertyTests. The following query selects all Authors of all Publications.

```
Publication()/hasAuthor
```

NodeTests and PropertyTests select not only the items of the specified type, but also the items of all subtypes and equivalent types, i.e., all types that are subsumed by the type of the test expression. This is an important difference to the semantics of general XPath expression. Determination of subsumption between classes or properties of an ontology is accomplished by existing OWL reasoners, we use the Pellet [10] reasoner for this.

3.2 FLWOR Expressions

Besides SWQLPath the main construct in SWQL are FLWOR expressions as in XQuery. These are comparable to SELECT-FROM-WHERE expressions in SQL. The query "Select all Publications, whose title contains the String 'Data Integration' and that were published after '1998'" could be expressed with SWQL as follows:

```
for $a in Publication()
where
  contains($a/hasTitle, "Data Integration")
  and $a/hasYear > 1998
return $a
```

The next query demonstrates a multi-way join and selects the name of all authors that have published at the VLDB conferences that took place after 1995 ordered by name:

```
for $a in Person(),
    $p in Publication(),
    $c in Conference()
where $c/hasYear > 1995
  and $c/hasName = "VLDB"
  and $p/isPublished = $c
  and $p/hasAuthor = $a
order by $a/hasName
return $a/hasName
```

The following query demonstrates aggregation and returns the number of available publications, where "I.M.Author" is on of the authors:

```
let $a := Publication()[hasAuthor = "I.M.Author"]
return count($a)
```

3.3 SWQL Constructors

Constructors allow to create new instances of objects and properties. An Object-Constructor creates an object of a specified class with a set of properties. The following example creates a new Article instance, with a title and publication year.

```
object Article {
   hasTitle { "Data on the Web" }
   hasYear { 2000 }
}
```

Because the results of SWQL queries are graphs that can be cyclic, in particular when constructed by recursive method/constructor calls, duplicate objects must be avoided. Therefore it is possible to optionally declare an id for a constructed object. This id is used during object construction to check if an object with this id has already been constructed. This skolem functionality prevents the creation of duplicate objects. The following example shows the previous query with using an id.

```
object Article "1234" {
   hasTitle { "Data on the Web" }
   hasYear { 2000 }
}
```

Constructors are implicitly validated against their ontology definitions, which guarantees that the resulting object is a valid instance of the type as specified in the ontology.

4 Using SWQL as mapping language

The SWQL constructors can be used together with user-defined functions to define mappings between different ontologies. A SWQL function can then take an instance of one ontology and constructs an instance of another ontology out of it. To demonstrate such a mapping the schema in Figure 5 should be mapped to the target schema in Figure 4. Both schemas represent objects in the address domain, they are heterogeneous in the used vocabulary and their structure and they are both cyclic.

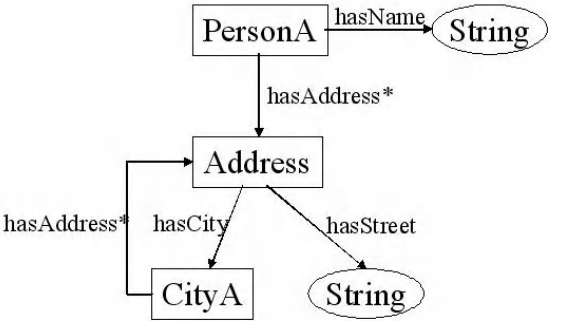


Fig. 4. The target schema

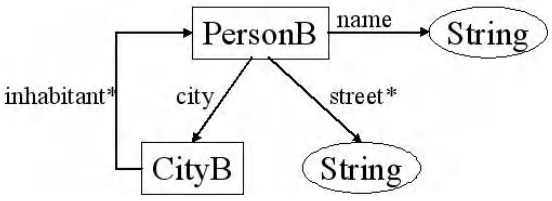


Fig. 5. The local schema

```
declare function createPersonA($p as PersonB)
   as PersonA
{
   object PersonA "{id($p)}" {
      hasName { $p/name }
      hasAddress {
         for $a in $p/street
         return createAddress($a, $p/hasCity)
      }
   }
}

declare function createAddress($s as xs:string,
                               $c as CityB )
   as Address
{
   object Address {
      hasStreet { $s },
      hasCity { createCityA($c) }
   }
}

declare function createCityA($c as CityB)
   as CityA
{
   object CityA "{id($c)}" {
      hasAddress {
         for $p in $c/inhabitant
         return createPersonA($p)/hasAddress
      }
   }
}
```

Fig. 6. Mapping from the local to the target schema

The mapping from such local schema instances to target schema instances can be seen in Figure 6. This mapping copies the name of the person and creates a new Address object for every street-city combination of the local schema. The hasAddress relationship of CityA to Address is realized by collecting all addresses of all inhabitants of CityB.

Because of the static typing feature of SWQL such a mapping can be checked for consistency, i.e., it can check that the resulting Article object really corresponds to the ontology definition. This allows to design and validate such mappings without testing it on the actual instances.

5 Related Work

RDF query languages [1] like RQL [11], RDQL [12] or SeRQL [13] are also able to query data that is related to an ontology, but only if the data is RDF. These languages do not provide constructors and user-defined functions or similar features that would allow to use them as mapping language for overcoming the heterogenity between different ontologies. The construction feature of SeRQL is essentially an update feature, as it simply adds additional triples to the triple store.

XQuery [5] is designed very generic, but it is not able to use ontologies directly. Also the XML tree data model of XQuery does not distinguish between the different semantics of nodes and edges in a graph data model.

OQL [14] the "Object Query Language" operates on a graph data model, and is able to construct new instances of existing classes. However, it is not able to create fully connected subgraphs with possible cyclic dependencies, but only instances referring to existing objects. This prevents the usage of OQL as mapping language between different graph structures.

6 Conclusion and Future Work

We have presented the SWQL query language, which is able to query arbitrary data sources on the basis of an OWL ontology. This query language offers amongst others, construction of new objects, user-defined functions and static type checking. This allows it to be used as ontology-based data integration tool, i.e., instances of one ontology can be transformed into instances of another and this mapping can be statically checked for consistency.

A full implementation of SWQL is developed and tested against XML and relational data sources. In the next step this implementation will be further stabilized and documentated for a first public release. This will hopefully produce additional feedback on the language as well as on the implementation in order to extend or polish the existent functionality.

Acknowledgments

This work is supported by the bmb+f in the SemIPort project.

References

1. Haase, P., Broekstra, J., Eberhart, A., Volz, R.: A comparison of rdf query languages. In: Proceedings of the Third International Semantic Web Conference, Hiroshima, Japan, 2004. (2004)
2. Dean, M., eds., G.S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A.: OWL Web Ontology Language Reference. http://www.w3.org/TR/owl-ref/ (2003) W3C working draft.
3. Connolly, D., van Harmelen, F., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A.: DAML+OIL Reference Description. http://www.w3.org/TR/daml+oil-reference/ (2001) W3C note.
4. Lassila, O., Swick, R.R.: Resource Description Framework (RDF) Model and Syntax Specification. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/ (1999) W3C Recommendation.
5. Boag, S., Chamberlin, D., Fernandez, M.F., Florescu, D., Robie, J., Siméon, J.: XQuery 1.0: An XML query language. http://www.w3.org/TR/xquery/ (2003) W3C working draft.
6. Biron, P.V., Malhotra, A.: XML Schema Part 2: Datatypes. http://www.w3.org/TR/xmlschema-2/ (2001) W3C recommendation.
7. Berglund, A., Boag, S., Chamberlin, D., Fernandez, M.F., Kay, M., Robie, J., Siméon, J.: XML path language (XPath) 2.0. http://www.w3.org/TR/xpath20/ (2003) W3C working draft.
8. Lehti, P., Fankhauser, P.: XML data integration with OWL: Experiences and challenges. In: Proceedings of the 2004 Symposium on Applications and the Internet (SAINT 2004). (2004) 160–170
9. Lehti, P., Shreshta, N., Hollfelder, S.: The Semantic Web Query Language SWQL. http://ipsi.fraunhofer.de/oasys/projects/semiport/ (2003)
10. Sirin, E.: Pellet OWL Reasoner. (http://www.mindswap.org/2003/pellet/)
11. Karvounarakis, G., Alexaki, S., Christophides, V., Plexousakis, D., Scholl, M.: RQL: A declarative query language for RDF. In: Proceedings of the 11th Intl. World Wide Web Conference (WWW2002)., Honolulu, Hawai, USA (2002)
12. Seaborne, A.: Rdql - a query language for rdf. http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/ (2004) W3C member submission 9 january.
13. Broekstra, J., Kampman, A.: Serql: An rdf query and transformation language. http://www.cs.vu.nl/ jbroeks/papers/SeRQL.pdf (2004) Draft.
14. : Odmg oql user manual. http://www.cis.upenn.edu/ cis550/oql.pdf (1998)

Modeling Views for Semantic Web Using eXtensible Semantic (XSemantic) Nets

R. Rajugan[1], Elizabeth Chang[2], Ling Feng[3], and Tharam S. Dillon[1]

[1] eXel Lab, Faculty of IT, University of Technology, Sydney, Australia
{rajugan, tharam}@it.uts.edu.au
http://exel.it.uts.edu.au
[2] School of Information Systems, Curtin University of Technology, Australia
Elizabeth.Chang@cbs.cutin.edu.au
[3] Faculty of Computer Science, University of Twente, The Netherlands
ling@ewi.utwente.nl

Abstract. The emergence of Semantic Web (SW) and the related technologies promise to make the web a meaningful experience. Yet, high level modeling, design and querying techniques proves to be a challenging task for organizations that are hoping utilize the SW paradigm for their industrial applications, which are still using traditional database techniques. To address such an issue, in this paper, we propose a view model for the SW (SW-View), to SW-enable traditional solutions. First we outline the view model, its properties and some modeling issues, followed by some discussions on modeling such views (at the conceptual level). We also provide a brief discussion on how this view model is utilized in the design and construction of materialized ontology views to support extraction of sub-ontologies.

1 Introduction

Many traditional database concepts and techniques have been transformed and adopted to new web application platforms, which are mainly based on core Object-Oriented (OO) principles. For example, works such as [2-4, 16, 25] are good examples in this direction. The emergence of Semantic Web (SW) [33] and the related technologies promise to make the web a meaningful experience and it is another step towards the next generation of Enterprise Information Systems (EIS). However, success of SW and its applications heavily depends on utilization and interoperability of well formulated Ontology bases (and traditional data) in an automated, heterogeneous environment. For example, utilization, integration and extraction of ontology bases in the context of EIS, where, enterprise vocabularies can be automatically extracted from various distributed sources and be used in one or more SW (or traditional) applications and e-services. One such scenario is shown in Fig. 1.

This creates the need investigate successful database technologies, such as views, in the context of SW, where (materialized) ontology views [37] can be used for; (a) ontology extraction, (b) ontology versioning (c) SW-enabling traditional data sources and (d) sub-ontology generation, in an industrial settings. However, unlike traditional database systems, high level modeling, design and querying techniques still proves to be a challenging task for SW paradigm. This is mainly due to the nature of ontology

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 936 – 946, 2005.
© Springer-Verlag Berlin Heidelberg 2005

bases and views, where, definitions and querying have to be done at high-level abstraction [30, 37]. Such a high-level view models can also be utilized in SW paradigm and also support and co-exist with existing traditional database architecture and/or enterprise transactional systems. A detailed discussion on the differences between the traditional database and SW technologies can be found in our work [26].

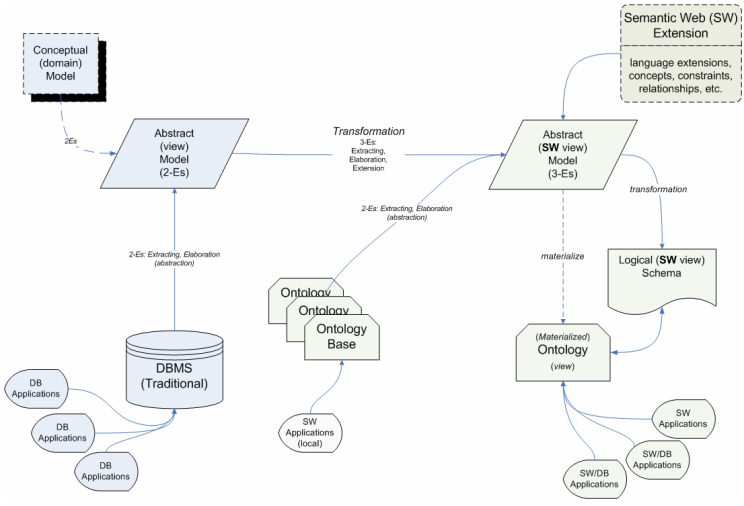


Fig. 1. Databases and Ontology bases in EIS architecture (context diagram)

Conversely, Semantic Web directives are still at its infancy in areas such as data organization, meta-data models and query languages. But there is an exponential growth in new research directions in SW applications. These applications range from SW enabled traditional enterprise meta-data repositories (Fig. 1) to time-critical medical information and infectious decease classification databases. For such vast ontology bases to be successful in a distributed environment, the preliminary design and engineering of such ontology bases should follow a strict software engineering discipline [28]. Furthermore, supporting technologies for ontology engineering such as data extraction, integration and organization have be matured to provide adequate modeling and design mechanism to build, implement and maintain successful ontology bases. For such purpose, Object-Oriented (OO) paradigm seems to be an ideal choice as it has been proven in many other complex applications and domains [12, 18].

To address such an issue, in this paper, we propose a view model for SW, to support ontology views (Fig. 1). In contrast to SW language specific views (e.g. RDF [32] /RDF-S), the proposed view model is defined using a high-level modeling OO language that is capable of modeling ontologies and sub-ontologies (for e.g. XSemantic nets [14] or OMG's UML [23] or Ontology Web Language (OWL) [31]). Our main aim here is to "re-use" and "share" of view definitions among multiple implementation paradigms and frameworks (Fig. 1), namely; (a) (traditional) database systems, (b) database applications and (c) SW applications, as only the use, encoding,

relationships and meta-data may change between these platforms in (a) to (c) above (*see* the example case study description in section 4). Thus we provide view definitions at the highest level of abstraction (i.e. conceptual level) which enables us to transform and map one view definition to a specific platform (i.e. (a), (b) or (c)), at the required level of abstraction (i.e. conceptual, schema or instance). Here, our focus is mainly on SW paradigm.

The rest of this paper is organized as follows. In section 2, we present some of the early work done in Semantic Web view models. Section 3 describes our work (including a brief outline on how our view model is applied in the MOVE system), followed by section 4, where we present an illustrative case study example to highlight some of our view model characteristics. Section 5 concludes the paper with some discussion on our future research directions.

2 Related Work

We can group the existing view models into four categories, namely; (a) classical (or relational) views [11, 13], (b) Object-Oriented (OO) view models [5, 22], (c) semi-structured (namely XML) view models [1, 10, 25] and (d) view models for SW. An extensive set of literature can be found in both academic and industry forums in relation to various view related issues such as (i) models, (ii) design, (iii) performance, (iv) automation and (v) turning/refinement, mainly supporting the 2-Es; data Extraction and Elaboration (with and some research directions towards 3-Es, i.e. 2-Es and data Extension). A comprehensive discussion on existing view models can be also found in [25, 26]. Here, we focus only on view models for SW.

In related work in Semantic Web (SW) [34] paradigm, some work has been done in views for SW [29, 30], where the authors proposed a view formalism for RDF document with support for RDF [32] schema (using a RDF schema supported query language called RQL). This is one of the early works focused purely on RDF/SW paradigm and has sufficient support for logical modeling of RDF views. The extension of this work (and other related projects) can be found at [21]. RDF is an object-attribute-value triple, where it implies object has an attribute with a value [15]. It only makes intentional semantics and not data modeling semantics. Therefore, unlike views for XML, views for such RDF (both logical and concrete) have no tangible scope outside its domain. In related area of research, the authors of the work propose a logical view formalism for ontology [36, 37] with limited support for conceptual extensions, where materialized ontology views are derived from conceptual/abstract view extensions.

Another area that is currently under development is the view formalism for SW Meta languages such as OWL. In some SW communities, OWL is considered to be a conceptual modeling language for modeling ontologies, while some others consider it to be a crossover language with rich conceptual semantics and RDF like schema structures [36]. It is outside the scope of this paper to provide argument for or against OWL being a conceptual modeling language. Here, we only highlight one of view formalism that is under development for OWL, namely views for OWL in the "User Oriented Hybrid ontology Development Environments" [19] project.

3 Our Work: SW-View Model

In this paper, we propose a layered view model for the SW paradigm (SW-view). Initially, we proposed a layered view model in our work for semi-structured data (namely XML) [25], and here we extend the model for SW paradigm.

In work with XML, we provided clear distinction between conceptual, logical and document levels views, as in the case of data engineering, there exists a need to clearly distinguish these levels of abstractions. But in the case of SW (e.g. ontologies), though there exists a clear distinction between conceptual and logical models/schemas, the distinction between the logical (or schema) level and document (or instance) level trends to overlap due to the nature of ontology bases, where concepts, relationships and values may present mixed sorts such as schemas and values [38].

Therefore, in the SW-view model, we provide a clear distinction between conceptual and logical views, but depending on the application, we allow an overlap between logical and document views. This is one of the main differences between the XML views and the SW-views. To our knowledge, other than our work, there exist no research directions that explore the conceptual and logical view formalism for the Semantic Web (SW) paradigm. This notion of SW-view model has explicit constraints and an extended set of conceptual operators to support ontology Extraction Methodology (OEM) [35, 37, 38].

3.1 Conceptual Views

In the layered view model, the conceptual views are views that are defined at the conceptual level with conceptual level semantics using a higher-level modeling languages such as UML [23] or XSemantic nets [14, 27]. Here, we use XSemantic nets. To understand the SW-view and its application in constructing ontology views, it is imperative to understand its concept and its properties. It should be noted here that, though there can be more elaborated definitions are possible depending on the application domain, here we provide a simplified generic conceptual view definition that can be easily applied.

***Definition 1*:** A **conceptual view** $\boldsymbol{V^c}$ is a 4-ary tuple $\boldsymbol{V^c} = (V^c_{name}, V^c_{obj}, V^c_{rel}, V^c_{constraint})$, where V^c_{name} is the name of the XML conceptual view V^c, V^c_{obj} is a set of objects in V^c, V^c_{rel} is a set of object relationships in V^c, and $V^c_{constraint}$ is a set of constraints associated with V^c_{obj} and V^c_{rel} in V^c.

***Definition 2*:** Let $C = (C_{name}, C_{obj}, C_{rel}, C_{constraint})$ denote a context which consists of a context name C_{name}, a set of objects C_{obj}, a set of object relationships C_{rel}, and a set of constraints associated with its objects and relationships $C_{constraint}$. Let $\hat{\lambda}$ be a set of conceptual operators. $\boldsymbol{V^c} = (V^c_{name}, V^c_{obj}, V^c_{rel}, V^c_{constraint})$ is called a *valid conceptual view of the context* $\boldsymbol{C}$, if and only if the following conditions satisfy;

1. For any object $\forall o \in V^c_{obj}$, there exist objects $\exists o_1, \ldots, o_n \in C_{obj}$, such that $o = \lambda_{1\ldots}\lambda_m (o_1, \ldots, o_n)$ where $\lambda_{1\ldots}\lambda_m \in \hat{\lambda}$. That is, o is a newly derived object from existing objects $o_1, \ldots, o_n$ in the context via a series of conceptual operators [24, 38] $\lambda_{1,\ldots}\lambda_m$ like select, join, etc.

2. For any constraint $\forall c \in V^{c}_{constraint}$, there exists a constraint $\exists c' \in C_{constraint}$ or a new constraint *c''* constraints associated with V^{c}_{obj} or V_{rel}.
3. For any hierarchical relationship $\forall r^{h} \in V^{c}_{rel}$, there *does not exist* a relationship between one or more and V^{c}_{obj} and C_{obj}.
4. For any association relationship/dependency relationships $\forall r^{a} \in V^{c}_{rel}$, there *may exist a relationship between one or more* V^{c}_{obj} *and* C_{obj}.

The term ***context*** refers to the domain that interests an organization as a whole. It implies a meaningful collection of objects (or concepts), relationships (both structural and semantic) among these objects, as well as some constraints associated with the objects and their relationships, which are relevant to its applications. The following sections briefly address some of the unique characteristics of conceptual views; (i) conceptual operators, (ii) some modeling issues and (iii) the descriptive constraint specification for conceptual views using XSemantic nets.

3.2 Conceptual Operators

A Context is presented in XSemantic nets using modeling primitives like object (node), attribute (simple node), relationship (directed edges) and constraint in this study. To enable the construction of a valid conceptual view from a context, we introduced the notion of *conceptual operator (*$\hat{\lambda}$*)*[24]. These operators are grouped into set operators, namely union, difference, intersection, Cartesian product and unary operators namely projection, rename, restructure, selection and joins, and can facilitate systematic construction of conceptual views from context. These conceptual operators can be easily transformed into query segments, user-defined functions and/or procedures for implementation. By doing so, they help the modeler to capture view construct at the abstract level without knowing or worrying about query/language syntax. The set of binary and unary operators provided here is a complete or basic set; i.e. other operators, such as division operator and compression operator [38] can be derived from these basic set of operators.

3.3 Modeling Conceptual Views

In this paper, to model conceptual views, we use XSemantic nets. Other modeling notations used to model conceptual views can be found in [38]. XSemantic net provides a well defined, rich semantics to visually model a given domain into needed level of abstraction [14]. In the case of Ontology engineering, XSemantic nets provide rich collection of OO concepts and elements, namely; (i) classes (similar to concepts in ontology), (ii) attributes (iii) relationships (and cardinality constraints) between classes, (iv) relationships (and cardinality constraints) between class and its attributes and (v) a rich set of constraints (*see* section 3.4 below). Some of the XSemantic net notations are given Fig. 2 and an illustrative case study example model is given in section 4.

Base on the V^{c} definition 1 above, in XSemantic nets, V^{c}_{obj} are shown using (simple/complex) nodes, V^{c}_{rel} using edges and $V^{c}_{constraint}$ using constraints defined over (a set of) node/(s) and (a set of) edges.

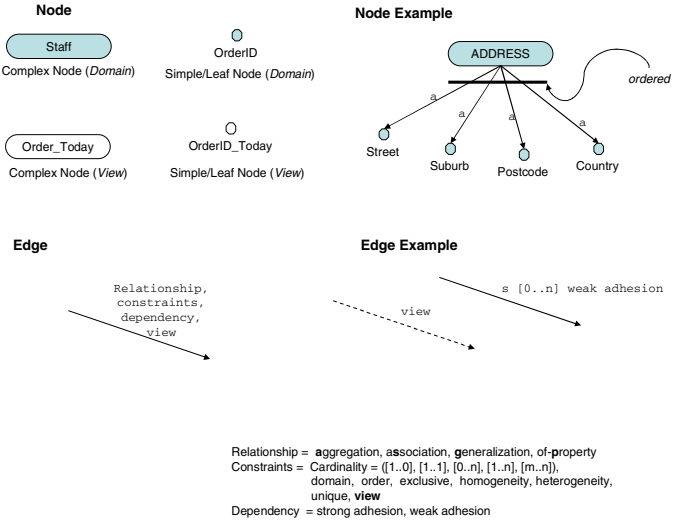


Fig. 2. XSemantic net notations

3.4 Modeling Conceptual View Constraints

One of the main differences between traditional data modeling and modeling ontologies, is the constraint specification. In ontology modeling, it requires a rich set of high-level constraint specification mechanism. In the case of views (both traditional data and ontology views), it is usually specified by the data/query language in which they are defined. For example, in relational model, views are defined using SQL and a limited set of constraints can be defined using SQL[11, 13], while in Object-Relational and OO models, views have similar constraints but they are more extensive and explicit due to the nature of the data model. The views here are constructed and specified by using DBMS specific (such as OQL[7]) and/or external languages (such as C++, Java or O_2C[5]). It is a similar situation in views for semi-structured data paradigm, where rich set of view constrains are defined using languages such as OQL based LOREL [6, 17]. Today, in the case of Ontology engineering (and in ontology views), this is still holds true, where constraints are specified using programming modules than at the schemata and/or logical level. In doing so, the constraints are implicit and mostly accessible only at runtime of the system and not at the modeling and/or design time.

But the work by authors of [10] provides some form of higher-level view constraints (under ORA-SS model) for XML views, while the work in [30] provides some form of logical level view constraints to be defined in views for in SW/RDF paradigm. Here, for our view formalism, we look into using XSemantic net as the (visual) constraint specification language.

3.5 Constraint Specification Using XSemantic Net

XSemantic net is a modified semantic network model, to model data domains under the OO paradigm [14, 27]. In XSemantic nets, due to its structural similarity to semi-structured data (e.g. XML, RDF etc.), most data/schema specific constraints are build-in to the model. There exist no need to have additional (textual) constraint specification language (e.g. such as OCL). These constraints are grouped into three categories [14], namely; (a) constraints over an edge, (b) constraints over a set of edges and (c) constraints over an edge. In addition, further constraints can be defined for conceptual views including; (i) domain constraints (range of values, min, max, pattern etc), (ii) constructional contents (set, sequence, bag, ordered-set), (iii) ordering (iv) explicit homogenous composition/heterogeneous compositions, (v) adhesion and/or dependencies (vi) exclusive disjunction and many more. Specifying these constraints in XSemantic nets (or UML/OCL) for conceptual views is similar to that of stored domain object constraints. The notations used in XSemantic net are given in Fig. 2. In section 4, we demonstrate constraint specification using XSemantic nets using some case study examples.

3.6 Conceptual Views on the MOVE System

Here, we briefly discuss how SW-views can be applied in the Materialized Ontology View Extractor (MOVE) system [36] for ontology extraction. The MOVE system was initially proposed by Wouters et al. [35-37], for the construction of *optimized* materialised ontology views, with emphasis on automation and quality of the views generated. The MOVE view process includes model and design of conceptual views with the utilization of restricted conceptual operators in deriving materialized ontology views. Some of the restricted view operators (derived from one or more SW-view conceptual operators) include [37, 38]; (a) synonymous rename (2) selection and (3) compression. A detailed discussion in this topic can be found in [38] and detailed work on MOVE can be found in [35-37].

***Definition 3*:** [38](Informal) A Strict Semantic Web View (or Ontology View) is a materialized SW-view that is derived from an ontology (called the base ontology). The derivation can consist of any (combination) of the following operations; synonymous rename, selection and compression.

4 An Illustrative Case Study Example

To help illustrate our concepts, we conduct a real-world case study in a fictitious global logistic company called LWC & e-Solutions Inc., e-Sol in short. The e-Sol Inc. aims to provide logistics, warehouse, and cold storage space for its global customers and collaborative partners. The e-Sol solution includes a standalone and distributed Warehouse Management System (WMS/e-WMS), and a Logistics Management System (LMS/e-LMS) on an integrated e-Business framework called e-Hub [8] for all inter-connected services for customers, business customers, collaborative partner companies, and LWC staff (for e-commerce B2B and B2C). Some real-world applications of such company, its operations and IT infrastructure can be found in [8, 9, 20].

Here, we use this system as the base to model and integrate (using views) various (traditional) databases, ontology bases and other sub-ontology vocabularies used at various customer and collaborative partner locations (Fig. 1).

In e-Sol, due to the business process, data semantics have to be in different formats (ontology bases, databases, XML and vocabularies) to support multiple systems, customers, warehouses and logistics providers. Also, data have to be duplicated at various points in time, in multiple databases, to support collaborative business needs. In addition, since new customers/providers join the system (or leave), the data formats has to be dynamic and should be efficiently duplicated without loss of semantics. This presents an opportunity to investigate how to integrate and utilize various customers' and collaborative partners' (data and ontology) bases for mutual benefit and for SW applications. The following examples highlight some of the conceptual views developed for the e-Sol. Note: It should be note that, the examples and the figures given for the e-Sol are demonstration purpose only and do not provide the complete ontology base model of the system.

Example 1: "`staff`", "`order`", and "`customer`" can be some of the context examples in the e-Sol system.

Example 2: "`processed-order`" and "`overdue-order`" are two contrasting conceptual views in the context of "`order`" of the e-Sol system.

Example 3: "`Warehouse-Manager`" is a valid conceptual view, named in the context of "`Staff`". It is constructed using the conceptual SELECT operator [24], which can be shown as;

$$\sigma_{\texttt{warehouse-Staff.Role="manager"}}(\texttt{Users}).$$

Example 4: Similarly, the conceptual view of name "`Site-Manager`" in the given context "`Staff`".

Example 5: In the case of conceptual view "`Warehouse-Manager`" (Fig. 3), we indicate the unique `staffID` using the unique constraint.



Fig. 3. Unique Constraint

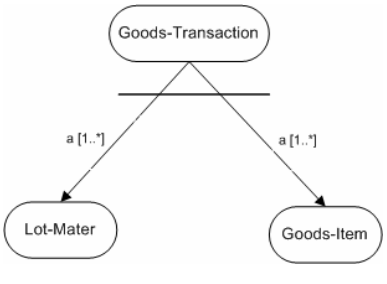


Fig. 4. Ordered composition in e-Sol

Example 6: In real-world, composite objects being in an aggregation with one or more sub-objects (Fig. 4), they also can be in a pre-defined order. This signifies an important OO concept, *ordered composition*.

Example 7: In the case of conceptual views "`Lot-Movement`", the exclusive disjunction between `Internal-Lot-Movement` (stored goods change owners) and `External-Lot-Movement` (goods shipped outside the warehouse) can be shown as in Fig. 5.

Example 8: *One* `Goods-Type` composes of *one-or-more* `Goods-Sub-Type`/(s) and one `Goods-Item` is associated with *one-or-more* `Goods-Sub-Type`/(s), as shown as in Fig. 6.

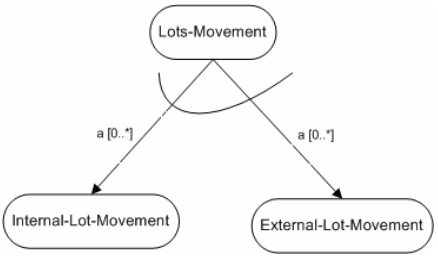


Fig. 5. Exclusive disjunction constraint

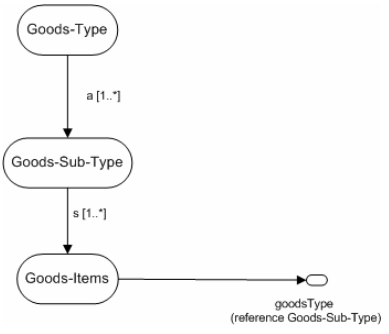


Fig. 6. A cardinality constraint example

Example 9: In the case of conceptual views "`Warehouse-Manager`" and "`Warehouse-Staff`", in the context of "`Staff`", we indicate the adhesion relationship, as shown as in Fig. 7.

Example 10: In the case of conceptual view "`Income`", the dependency (constraint) relationships shown in Fig. 8 hold true.

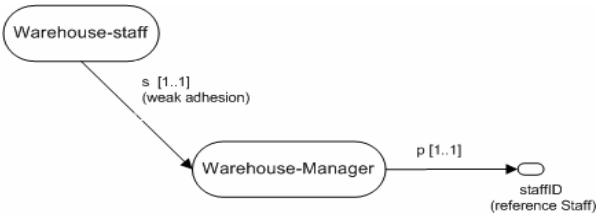


Fig. 7. Dependency / adhesion constraint

Example 11: A compression of elements indicates that those elements are replaced by a single element in the ontology view [38]. The element itself can be a new element, but it will not provide additional semantic information (compared to the base ontology). The compression operator constituted of one or more of unary operations combined in sequence.

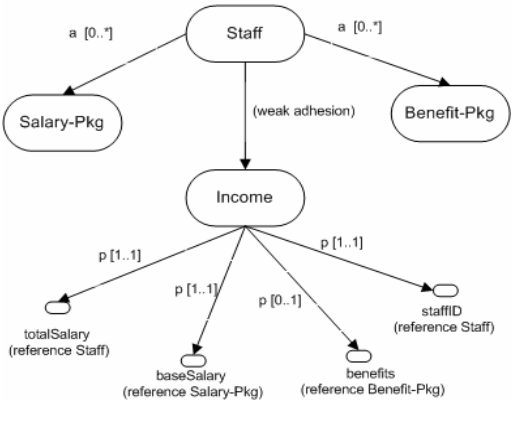


Fig. 8. A complex dependency constraint

5 Conclusion and Future Work

Views have proven to be very useful in databases and here, we discussion on an abstract view model for SW (SW-view). First, we described the opportunities and challenges for utilizing SW technologies for EIS and databases. Then we briefly provided some arguments for our SW-view model and discussed its properties, definitions and modelling aspects. We also briefly showed how such view model is applied in the MOVE system for ontology extraction. Finally, we presented a practical walkthrough of the view model using an industrial case study example.

For future work, some further issues deserve investigation. First, the investigation of a formal mapping approach to conceptual view constraints, to automate the view constraint model transformation between the SW-view model and one or more SW language (such as RDF and OWL schema) constraints. Second, the automation of the mapping process between conceptual operators to various SW (high-level) query language expressions (e.g. RDQL) with emphasis on performance.

References

1. S. Abiteboul, "On Views and XML," Proc. of the 18th ACM PODS '99, USA, 1999.
2. S. Abiteboul, et al., "Active Views for Electronic Commerce," Proc. of the 25th Int. Conf. on VLDB, Edinburgh, Scotland, 1999.
3. S. Abiteboul, O. Benjelloun, I. Manolescu, T. Milo, and R. Weber, "Active XML: A Data-Centric Perspective on Web Services," BDA, 2002.
4. S. Abiteboul, O. Benjelloun, I. Manolescu, T. Milo, and R. Weber, "Active XML: Peer-to-Peer Data and Web Services Integration," Proc. of the 28th Int. Conf. on VLDB, HK, China, 2002.
5. S. Abiteboul and A. Bonner, "Objects and Views," ACM SIGMOD Record, Proc. of the Int. Conf. on Management of Data (ACM SIGMOD '91), 1991.
6. S. Abiteboul, J. Quass, J. McHugh, J. Widom, and J. Wiener, "The Lorel Query Language for Semistructured Data," *Int. Journal on Digital Libraries*, vol. 1, pp. 68-88, 1997.
7. R. G. G. Cattell, et al., "The Object Data Standard: ODMG 3.0," Morgan Kaufmann, 2000, pp. 300.
8. E. Chang, et al., "A Virtual Logistics Network and an e-Hub as a Competitive Approach for Small to Medium Size Companies," 2nd Int. Human.Society@Internet Conf., Seoul, Korea, 2003.
9. E. Chang, et al., "Virtual Collaborative Logistics and B2B e-Comm.," e-Business Conf., NZ, 2001.
10. Y. B. Chen, T. W. Ling, and M. L. Lee, "Designing Valid XML Views," Proc. of the 21st Int. Conf. on Conceptual Modeling (ER '02), Tampere, Finland, 2002.
11. C. J. Date, *An introduction to database systems*, 8th ed. New York: Pearson/Addison Wesley, 2003.
12. T. S. Dillon and P. L. Tan, *Object-Oriented Conceptual Modeling*: Prentice Hall, Australia, 1993.
13. R. Elmasri and S. Navathe, *Fundamentals of database systems*, 4th ed. New York: Pearson/Addison Wesley, 2004.
14. L. Feng, E. Chang, and T. S. Dillon, "A Semantic Network-based Design Methodology for XML Documents," *ACM Transactions on Information Systems (TOIS)*, vol. 20, No 4, pp. 390 - 421, 2002.
15. L. Feng, E. Chang, and T. S. Dillon, "Schemata Transformation of Object-Oriented Conceptual Models to XML," *Int. Journal of Computer Systems Science & Engineering*, vol. 18, No. 1, pp. 45-60, 2003.

16. L. Feng and T. S. Dillon, "An XML-Enabled Data Mining Query Language XML-DMQL," *Int. Journal of Business Intelligence and Data Mining*, 2005.
17. R. Goldman, J. McHugh, and J. Widom, "From Semistructured Data to XML: Migrating the Lore Data Model and Query Language," Proc. of the 2nd Int. Workshop on the Web and Databases (WebDB '99), Philadelphia, Pennsylvania, 1999.
18. I. Graham, A. C. Wills, and A. J. O'Callaghan, *Object-oriented methods : principles & practice*, 3rd ed. Harlow: Addison-Wesley, 2001.
19. HyOntUse, "User Oriented Hybrid Ontology Development Environments, (http://www.cs.man.ac.uk/mig/projects/current/hyontuse/)," 2003.
20. ITEC, "iPower Logistics (http://www.logistics.cbs.curtin.edu.au/)," 2002.
21. KAON, "KAON Project (http://kaon.semanticweb.org/Members/rvo/Folder.2002-08-22.1409/Module.2002-08-22.1426/view)," 2004.
22. W. Kim and W. Kelly, "Chapter 6: On View Support in Object-Oriented Database Systems," in *Modern Database Systems*: Addison-Wesley Publishing Company, 1995, pp. 108-129.
23. OMG-UML™, "UML 2.0 Final Adopted Specification (http://www.uml.org/#UML2.0)," 2003.
24. R.Rajugan, E. Chang, T. S. Dillon, and L. Feng, "A Layered View Model for XML Repositories & XML Data Warehouses," The 5th Int. Conf. on Computer and Information Technology (CIT '05), Shanghai, China, 2005.
25. R.Rajugan, E. Chang, T. S. Dillon, and L. Feng, "A Three-Layered XML View Model: A Practical Approach," 24th Int. Conf. on Conceptual Modeling (ER '05), Klagenfurt, Austria, 2005.
26. R.Rajugan, E. Chang, T. S. Dillon, L. Feng, and C. Wouters, "Modeling Ontology Views: An Abstract View Model for Semantic Web," 1st Int. IFIP/WG 12.5 Working Conf. on Industrial Applications of Semantic Web (IASW '05), Jyvaskyla, Finland, 2005.
27. R.Rajugan, E. Chang, L. Feng, and T. S. Dillon, "Semantic Modelling of e-Solutions Using a View Formalism with Conceptual & Logical Extensions," 3rd Int. IEEE Conf. on Industrial Informatics (INDIN '05), Perth, Australia, 2005.
28. P. Spyns, R. Meersman, and J. Mustafa, "Data Modeling Versus Ontology Engineering," SIGMOD, 2002.
29. R. Volz, D. Oberle, and R. Studer, "Implementing Views for Light-Weight Web Ontologies," Seventh Int. Database Engineering and Applications Symposium (IDEAS'03), Hong Kong, SAR, 2003.
30. R. Volz, D. Oberle, and R. Studer, "Views for light-weight Web ontologies," Proc. of the ACM Symposium on Applied Computing (SAC '03), USA, 2003.
31. W3C-OWL, "OWL: Web Ontology Language 1.0 reference (http://www.w3.org/2004/OWL/)," W3C, 2004.
32. W3C-RDF, "Resource Description Framework (RDF), (http://www.w3.org/RDF/)," 3 ed: The World Wide Web Consortium (W3C), 2004.
33. W3C-SW, "(http://www.w3.org/2001/sw/)," W3C, 2005.
34. W3C-SW, "Semantic Web, (http://www.w3.org/2001/sw/)," W3C, 2005.
35. C. Wouters, T. S. Dillon, J. W. Rahayu, and E. Chang, "A Practical Walkthrough of the Ontology Derivation Rules," Database and Expert Systems Applications : 13th Int. Conf. (DEXA '02), Aix-en-Provence, France, 2002.
36. C. Wouters, T. S. Dillon, J. W. Rahayu, E. Chang, and R. Meersman, "Ontologies on the MOVE," 9th Int. Conf. on Database Systems for Advanced Applications (DASFAA '04), Jeju Island, Korea, 2004.
37. C. Wouters, T. S. Dillon, J. W. Rahayu, E. Chang, and R. Meersman, "A Practical Approach to the Derivation of a Materialized Ontology View," in *Web Information Systems,*. USA: Idea Group Publishing, 2004.
38. C. Wouters, R.Rajugan, T. S. Dillon, and J. W. Rahayu, "Ontology Extraction Using Views for Semantic Web," in *Web Semantics and Ontology*, USA: Idea Group Publishing, 2005.

On the Cardinality of Schema Matching

Avigdor Gal

Technion – Israel Institute of Technology,
Technion City, Haifa 32000, Israel
avigal@ie.technion.ac.il

Abstract. In this paper we discuss aspects of cardinality constraints in schema matching. A new cardinality classification is proposed, emphasizing the challenges in schema matching that evolve from cardinality constraints. We also offer a new research direction for automating schema matching to manage cardinality constraints.

1 Introduction

Matching concepts describing the meaning of data in heterogeneous distributed data sources (*e.g.*, HTML form tags and database and XML schemata) is one of the basic operations of data integration. Due to the cognitive complexity of this matching process [2], it has traditionally been performed by human experts (Web designers, database analysts, and even lay users, depending on the context of the application) [14,9]. As data integration has been made more automated, the ambiguity in concept interpretation, also known as *semantic heterogeneity*, has become one of the main obstacles to this process. For obvious reasons, manual concept reconciliation in dynamic environments (with or without computer-aided tools) is inefficient to the point of being infeasible, and so cannot provide a general solution. Introduction of the Semantic Web vision [1] and the shift towards machine-understandable Web resources, combined with the increasing utilization of Web services have underscored the importance of automatic matching between sets of elements, also known as *schema matching*.

One of the comparison dimensions of schema matching that has been mentioned in the literature is that of mapping cardinality. Mapping cardinality has been defined in [13] as follows: "the overall match result may relate one or more elements of one schema to one or more elements of the other, yielding four cases: $1:1$, $1:n$, $n:1$, $n:m$. In addition, each mapping element may interrelate one or more elements of the two schemas. Furthermore, there may be different match cardinalities at the instance level." Mapping elements' cardinality is considered *global cardinality* while the mapping within elements is considered *local cardinality*. Cardinality is a continuous topic for debate in the research community. Some argue that $1:1$ mappings are not realistic. Others claim that $n:m$ are not sufficiently expressive and put much of the matching burden on the user. In [13] the authors conclude that "[m]ore work is needed to explore more sophisticated criteria for generating local and global $n:1$ and $n:m$ mappings, which are currently hardly treated at all."

This work aims at clarifying several aspects of mapping cardinality. We offer a refined classification of mapping cardinality, discuss the available solution for each cardinality type and suggest a general methodology to be followed for optimal fulfillment

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 947–956, 2005.
© Springer-Verlag Berlin Heidelberg 2005

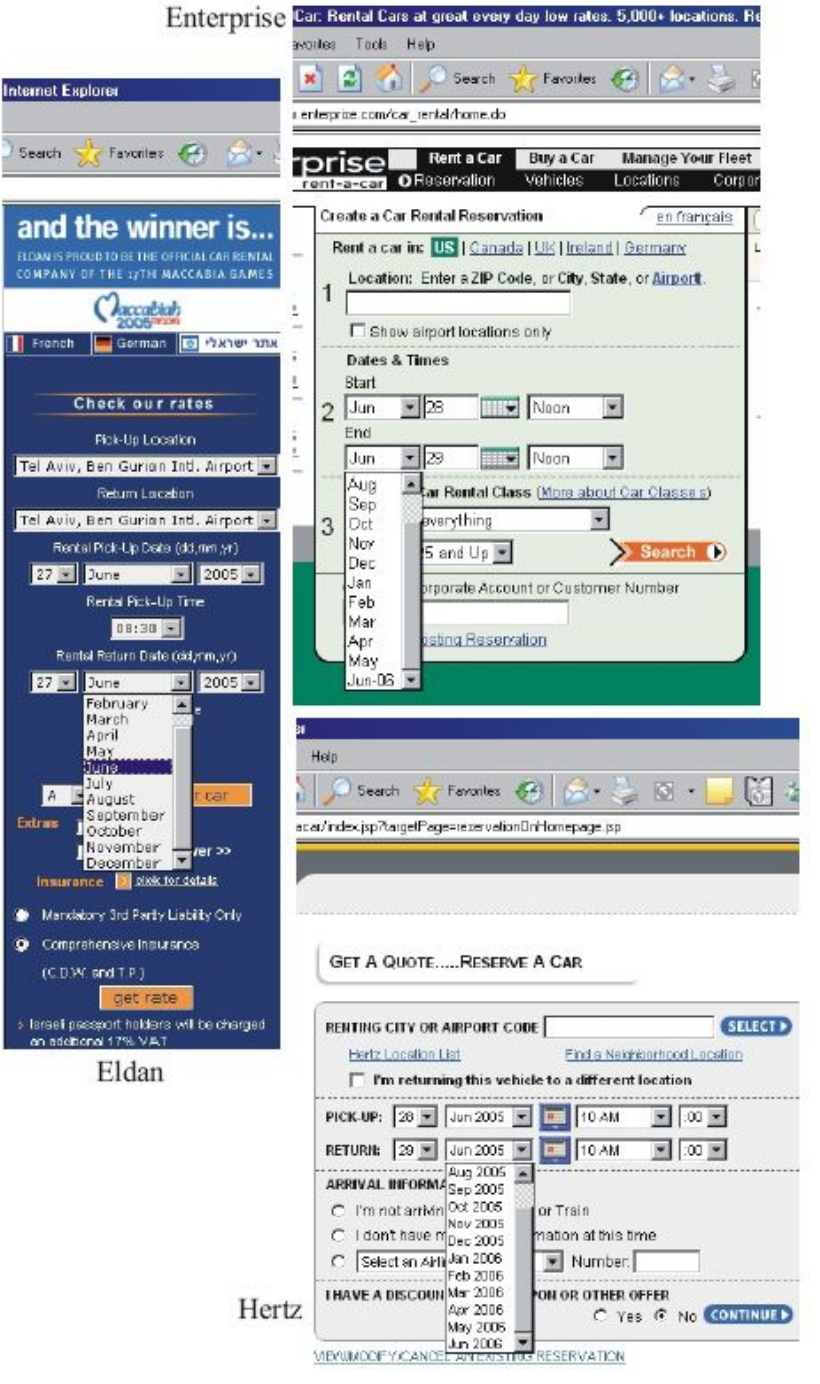


Fig. 1. Car rental case study

of the schema matching task. For demonstration purposes, we shall use examples based on Web forms. The filling of Web forms, which is typically intended for human beings, serves as a major challenge for schema matching heuristics. Web forms are schema-less and they lack standard presentation. Therefore, schema matching heuristics that work well in other applications (*e.g.*, relational databases and XML files) find it hard to match elements in this setting. As a concrete example, consider Figure 1. The figure introduces parts of Web forms of three car rental companies, namely Hertz, Enterprise, and Eldan. We focus on the date fields of pickup and return information. Hertz (on the bottom right part of the figure) has a combined date/year field with 13 possible values. Enterprise (top right) has a field for the month (with 13 values) and no field for year. The only reference to year is at the last value of the list of allowable values, with the value June-06. Finally, Eldan (left) has separate fields for month and year. When matching Hertz with Eldan, the cardinality constraint is $1 : n$. When matching Eldan with Enterprise, the cardinality constraint is $1 : 1$ (the year field of Eldan should not be mapped, though). It is not clear, however, what is the cardinality constraint when mapping Enterprise to Hertz.

The rest of the paper is organized as follows. We start by presenting a model for schema matching (Section 2), followed by our main observations on cardinality in this context (Section 3). Section 4 provides a refined cardinality classification and offers several strategies for satisfying cardinality constraints for this classification. Finally, we introduce a class of heuristics that identifies the appropriate cardinality given two schemata (Section 5) and provide concluding remarks (Section 6).

2 A Model for Schema Matching

As a basis for this work we next layout a model for schema matching and explicitly specify the set of assumptions we shall use throughout the paper. Let S_1 and S_2 be two schemata, defined using some data model (*e.g.*, relational or ontological), with n_1 and n_2 attributes, respectively. Attributes can be joint into *elements*, sets of attributes. The process of schema matching yields schema mapping(s), in which elements of S_1 are mapped onto elements of S_2.

Generally speaking, the process of schema matching is performed in two steps [3]. First, a degree of similarity is computed **automatically** for all element pairs (one element from each schema in each pair), using such methods as name matching, domain matching, and structure (such as XML hierarchical representation) matching. We refer to these heuristic methods as *schema matchers*. Recall that an element may consist of more than a single attribute.

As a second step, a single mapping is chosen to be the *best mapping*, using a *matching algorithm*. The best mapping is a mapping that optimizes some target function F, subject to matching constraints. For example, many schema matching tools aim at maximizing the sum (or average) of pair-wise weights of the selected elements. When deciding on a best mapping, a matching algorithm should decide which elements from one schema are to be mapped with elements of another schema. Also, the matching algorithm may decide that some elements do not satisfy some matching constraints (*e.g.*, minimal degree of similarity) and cannot be mapped.

COMA [3], OntoBuilder [5], Cupid [11], and other schema matching tools apply variations of this model in their matching process. Others (such as Prompt [4] and similarity flooding [12]) also apply this two step methodology, yet do not support a mode that provides the user with all pairwise element mappings. However, it can be expected that making their internal representation of attribute similarity measures available is feasible.

There are two essential differences between matchers and schema matching algorithms. First, a matcher is a heuristic evaluation of the similarity of two schema elements that can only be verified by empirical means. A matching algorithm aims at optimizing a target function, which correctness can be proven or bounded (in case the optimal algorithm has a non polynomial execution complexity). Second, a matcher is fully immersed in the application semantics. It may use subsumption, domain specific synonyms, *etc.* to determine the similarity measure to be assigned with a specific mapping. A matching algorithm receives as an input a "fully-baked similarity model" from the schema matcher and performs syntactic (from the application viewpoint) procedure to identify the best mapping.

A convenient data structure for modeling the matching problem is to view it as an undirected bipartite graph, $G = (X, Y, E)$, with a node set $V = X \bigcup Y$ representing elements, where X and Y denote the sides of the graph (each side representing one schema), and an edge set E. Weights $w : E \rightarrow \mathrm{R}^+$ are assigned with edges, representing the degree of similarity between elements, as provided by schema matchers. G does not have to be a complete graph. Threshold constraints may result in the elimination of some edges. Also, some matchers (*e.g.*, similarity flooding) present only partial pairwise similarity measures. Again, such constraints are interpreted as an incomplete graph.

A mapping in G is a subset of pair-wise edges of E. We denote a mapping by $M \subseteq E$. $F(M)$ represents the target function value of M. Each edge $e \in M$ is an *element mapping*.

Using the proposed data structure, the matching problem becomes a problem of selecting an optimal mapping (*i.e.*, a subset of E that optimizes F). Given a matching problem and a set of matching constraints, we denote by A_{best} the best known algorithm for solving the bipartite graph matching problem, given the matching constraints. We denote by $C(A_{best})$ the complexity of A_{best}. For example, when constraining the mapping to be $1:1$, the node set represents individual attributes, where the X node set contains all attributes of one schema ($|X| = n_1$) and the Y node set contains all attributes of the other schema ($|Y| = n_2$). A mapping in G is a subset of pair-wise **disjoint** edges of E. If F is defined to be weighted average, an efficient algorithm for identifying the best mapping in this case can be given as a variation of the weighted bipartite graph matching problem [6]. Such an algorithm has the complexity of $C(A_{best}) = O(n^3)$ [10],[1] where $n = \max(n_1, n_2)$. Other methods, such as stable marriage [8] can also serve in finding $1:1$ mapping, under different F definitions.

[1] Here we consider the complexity of the best *sequential* algorithm for finding a maximum weight mapping in a bipartite graph. Likewise, an alternative algorithm for this problem is presented in [7], and its time complexity is $O(n^{2.5} \log(nW))$, where W stands for the highest edge weight in the graph.

3 Application Cardinality and Algorithm Cardinality

We start our discussion with the following observation. Cardinality, as discussed in the literature, defines the nature of the matching algorithm, yet has been conceived to be part of the application specification. This observations stems directly from the differences between matchers and matching algorithms mentioned above. Since cardinality constraints are applied as part of the matching algorithm, it has no knowledge of the underlying application semantics (handled by the matcher). Some algorithms (*e.g.*, Cupid) have dodged the cardinality issue by declaring that the proposed heuristic can work with any (or at least several) cardinality constraints. Such avoidance is much easier to execute than say, avoiding the use of synonyms and the reason for that stems from the fact that the latter is to be used by the schema matcher while the former is part of the specifications of the matching algorithm. Another approach to tackling cardinality constraints was taken by OntoBuilder. A schema matcher, termed *Value*, compares the data types of schema elements to assess their similarity. Data types are preprocessed (called normalization in [5]) and reduced to atomic data types. Atomic data types can presumably be compared using a $1:1$ matching algorithm. This approach is an example of separating application from algorithm cardinality and attempting to adopt the former to the latter capabilities. To illustrate the difference between application and algorithm cardinality further, consider the matching of Enterprise with Hertz. The two car rental companies share the same business model, according to which car rental can be reserved up to one year in advance. The mirroring of this business rule on the Web form is done in different ways, though. While Hertz explicitly specify the allowed month/year combination, Enterprise implicitly interpreted the month to be within the next year with the exception of the last month, for which a year specification is added. The application cardinality for this field is actually $1:1$, despite the presumably very different representation.

What is application cardinality, then? we argue that application cardinality depends on the specific ontologies and even specific instantiations involved in the matching process and the inter-relationships between elements. As an example, consider once more the reservation forms of Figure 1. The example we gave at the end of Section 1 all relate to the same real-world entity (pickup or return month). When matching Hertz with Eldan, the cardinality constraint is $1:n$. When matching Eldan with Enterprise, the cardinality constraint is $1:1$. Cardinality cannot be considered closely related to entity types, then. It cannot be related to an application domain, either. It has to be judged on a case by case basis.

Moreover, there is most likely insufficient a-priori information about the application cardinality. To better understand this point, consider again the Web environment, and in particular Web form matching, as handled in OntoBuilder. In this setting, a matching problem is asymmetric. A matcher starts with a known schema (termed a *target ontology*), in which constraints can be easily specified. For example, consider the Hertz reservation site. The field "`I'm returning this vehicle to a different location`" has two possible valued (CHECKED and UNCHECKED). If this field is UNCHECKED, a field of "`Return Location`" cannot be mapped. In contrast, there is no apriori information on the new schema to be mapped (termed *candidate ontology*).

At a first glance, this may be considered "bad news" to our ability to correctly match schemata. However, using this observation as a starting point, one realizes that cardinality specification should not be propagated to the matching algorithm, but must remain at the level of matchers, and we will explain shortly how such a change in responsibility may work to the benefit of the schema matching process. Before that, however, we shall provide examples to the inter-relationships between schema matchers and matching algorithms. In particular, we show how a schema matcher can utilize a given matching algorithm to handle various application matching types.

4 From Application Cardinality to Algorithm Cardinality

Recall the typical classification of matching cardinality into either $1:1$ matching, $1:n$ matching, $n:1$ matching, and general $(n:m)$ matching [13]. Consider this classification as an algorithm cardinality. A matcher, using this or that technique, may determine that a specific application cardinality is needed. Next, we are concerned with the ability of transforming application cardinality into algorithm cardinality. We show a few examples of doing so, while we leave a full-fledged theory of the inter-relationships between application and algorithm cardinality to a future work.

We show two special cases of application cardinality and the methods for translating them into an algorithm that uses an undirected bipartite graph as the underlying data structure. Henceforth, we shall assume that $F(M) = \sum_{e \in M} k_e w(e)$, a weighted average where k_e is a parameter that can represent the relative importance of an edge e.

In the following examples, we differentiate between replications and decompositions, as follows. Consider, for example, a $1:n$ constraint. Such a constraint may indicate that a single attribute in one schema can be replicated to more than a single attribute in another schema (*e.g.*, a `Password` attribute in one schema vs. `Type Password` and `Retype Password` in another schema). Alternatively, such a constraint may indicate that an attribute in one schema is decomposed into several attributes in another schema (*e.g.*, `Name` in one schema is decomposed into `Given Name` and `Surname` in another schema). We denote the former a *replication* constraint and the latter a *decomposition* constraint. It is worth noting that replication constraint may be interpreted as $1:n$ global cardinality, joint with $1:1$ local cardinality while decomposition constraint may be interpreted as $1:n$ global cardinality, joint with $1:n$ local cardinality. For the latter, note that it is mandatory to match the whole set of n attributes in the second schema with the single attribute from the first schema. This is not the case in the classification of [13].

When a replication constraint is applied, matching decisions of individual attributes are independent of one another. Therefore, matching `Password` with `Type Password` is independent of the matching of `Password` with `Retype Password`. However, a decomposition constraint cannot be evaluated by some aggregation of matching of individual attributes. For example, consider the attribute `Name` and the attribute pair `Given Name` and `Surname`. Machine learning techniques are likely to rate the comparison of concatenation of values from `Given Name` and `Surname` against `Name`, higher than comparing each of the attributes independently. Therefore, decomposition constraints require the evaluation of elements as well as individual attributes.

4.1 Managing the Replication Constraint

Using the bipartite graph as a data structure, the following simple algorithm can be devised to support a $1:n$ replication constraint. Consider a matching constraint that allows an attribute in one schema to be replicated several times in another schema. Therefore, nodes on one side of the bipartite graph (say, the Y nodes) cannot have more than a single incidenting edge. Such a constraint does not apply to the other side of the graph (X nodes). Therefore, all one has to do is to identify the best edge incidenting upon each node that requires unique mapping. Let $v \in Y$ be a node in the graph and v_e be the set of all edges incident on v. The following simple algorithm can thus be applied.

Algorithm 1 $1:n$ with replication constraint

$S_e \leftarrow \emptyset$
for all $v \in Y$ **do**
  $S_e \leftarrow S_e \cup \{argmax_{v_e} w(e)\}$
end for
Return S_e

The complexity of Algorithm 1 is $C(A_{best}) = O(|E|) = O(n^2)$.

4.2 Managing the Decomposition Constraint

The bipartite graph can support duplication constraints by adding feasible attribute sets as elements (nodes) in the graph. Such enhancement entails, in many cases, higher complexity of the matching process. For example, if the matching constraint allows a single attribute in S_1 to be matched with up to 2 attributes in S_2, one needs to consider all possible pairs in a schema. Therefore, n_1 elements of S_1 are matched with $\binom{n_2}{2} = \frac{1}{2}n_2(n_2-1)$ elements of S_2, which increases the number of nodes in G to $|V| = O(n^2)$. This computation can be generalized to any (sufficiently small) constant c, constraining the number of attributes in an element. The complexity in this case is of $O(n^c)$.

Recall that constraints on the target schema are known. Therefore, a given set of individual constraints in the target schema determines the size of the matching problem as follows. The number of elements of S_2 is determined by the maximum number of attributes per element in any constraint on S_1. Then, individual constraints determine the total number of edges in the bipartite graph. As an example, assume that attribute A in S_1 can be mapped to up to two attributes from S_2, attribute B in S_1 can be mapped to up to three attributes from S_2 and all other attributes in S_1 can be mapped to a single attribute in S_2. The derived bipartite graph will have $\binom{n_2}{3}$ elements in S_2 (which follows from B's constraint on the maximum number of attributes in an element). Attribute A will have only $\binom{n_2}{2}$ edges, not connecting with elements of three attributes. Similarly, all other attributes will have n edges.

4.3 $n:m$ Mappings

Generally speaking, $n:m$ mappings provide, in and by itself, no effective constraints on the matching result. Combined with other parameters (such as thresholds), element mappings may be eliminated. Such general cardinality results in an increased manual effort, mainly for the elimination of redundant element mappings from the algorithm outcome.

When replacing single attributes with elements, traditional algorithms for solving matching problems in bipartite graphs can no longer ensure unique attribute selection. Therefore, two elements that contain the same attribute A can be chosen as part of a mapping, which means that A no longer has a unique mapping. Nevertheless, certain $n:m$ constraints can be supported by the bipartite graph data structure. As an example, consider a constraint enforcing each attribute in one schema to be mapped uniquely to a single combination of attributes. Therefore, if `Name` in S_1 is mapped to the combination of `Given Name` and `Surname` in S_2, no other attribute in S_1 can be mapped to this combination (although a different combination of `Surname` and `OfficeNumber` in S_2 may be the appropriate combination for `InitialPassword` in S_1). This special case can fall into the category of $1:1$ global cardinality, according to the classification of [13].

Such a constraint is mapped into the data structure in the following way. The X set of nodes represent legal elements of one schema. The Y set of nodes represent legal elements of the other schema. We can then apply an algorithm for solving the weighted bipartite graph matching problem in this graph. The complexity of this algorithm can be defined in terms of attributes as followed.

$$C(A_{best}) = O\left(|V|^3\right) = O\left((n^c)^3\right) = O\left(n^{3c}\right).$$

5 Determining Application Cardinality

So far, we have separated application cardinality from algorithm cardinality and shown several examples of translating application cardinality to known matching algorithms. The main problem still remains that of determining application cardinality. We now offer a direction for a new type of heuristics to determine the appropriate application cardinality. In what follows we assume that cardinality constraints are not known apriori, even for the target schema. For example, given a target schema with attribute `Name`, it is known whether it should be mapped to one or more attributes in the candidate schema.

The input of this heuristic is the same input as that of matchers, *i.e.*, two schemata. The output, which serves as an input to the matching algorithm, is a set of elements of each schema and the similarity measure among elements. The difference from the output in common practice is that the elements for this step are not determined in advance, but rather decided by the matcher during run-time. We emphasize this specific aspect, since it allows the matcher to combine various cardinalities for various parts of schemata. Some parts of schemata may be constrained to be $1:1$, while for other it may be constrained to be $1:n$, and so forth.

As an example, one can consider an iterative approach, in which a matcher assumes first that a 1 : 1 cardinality is enforced. In the presence of such a constraint, the matcher computes all pair-wise weights. It can next evaluate the quality of pair-wise mappings and decide (using some form of a threshold) that some attributes are poorly matched under 1 : 1 cardinality constraints. Such attributes can be assumed to have a different cardinality constraint, say 1 : 2, in which an attribute from one schema can be mapped to at most two attributes in another schema. Once a new cardinality constraint was set, the process repeats itself until the matcher is satisfied with the results, at which point the elements are transferred to the matching algorithm, along with the appropriate algorithm cardinality to generate the best mapping.

The details of such a heuristic are beyond the scope of this paper. However, considering the example of Figure 1 again, when matching Hertz with Eldan, the first iteration of 1 : 1 matching may yield poor results for the matching with month or year in the Eldan schema. Then, in the following iteration, if we assume that the only attributes that were left unmatched are those of the combined month/year of Hertz and the month and year of Eldan, we would be able to match, using 1 : 2 matching, the correct attributes.

6 Conclusions

In this work we have discussed schema matching cardinality. We have provided a model for schema matching and provided a reasonable justification for the separation of application and matching cardinality. Then, we have proposed a template for a set of heuristics that can iteratively determine the application cardinality for specific attributes in a schema.

As with any heuristic, heuristics of this type need to be empirically validated, to examine their limitations and understand better in which scenarios they will work best. In addition, the pursue of further understanding of the matching process, and especially in the Web environment, is a pressing matter. Without sufficient tools for automatic schema matching, the vision of the semantic Web and other models that aim at automatic transfer of information, free from manual involvement to the extent possible, will diminish.

Acknowledgments

Many thanks to Carmel Domshlak for useful discussions. The work was partially supported by two European Commission 6^{th} Framework IST projects, QUALEG and TerreGov, and the Fund for the Promotion of Research at the Technion.

References

1. T. Berners-Lee, J. Hendler, and O. Lassila. The semantic Web. Scientific American, May 2001.
2. B. Convent. Unsolvable problems related to the view integration approach. In Proceedings of the International Conference on Database Theory (ICDT), Rome, Italy, September 1986. In Computer Science, Vol. 243, G. Goos and J. Hartmanis, Eds. Springer-Verlag, New York, pp. 141-156.

3. H.H. Do and E. Rahm. COMA - a system for flexible combination of schema matching approaches. In Proceedings of the International conference on very Large Data Bases (VLDB), pages 610621, 2002.
4. N. Fridman Noy and M.A. Musen. PROMPT: Algorithm and tool for automated ontology merging and alignment. In Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000), pages 450455, Austin, TX, 2000.
5. A. Gal, G. Modica, H.M. Jamil, and A. Eyal. Automatic ontology matching using application semantics. AI Magazine, 26(1), 2005.
6. Z. Galil. Efficient algorithms for finding maximum matching in graphs. ACM Computing Surveys, 18(1):2338, March 1986.
7. U. Guntzer, W-T. Balke, and W. Kieling. Optimizing multi-feature queries in image databases. In Proceedings of the Twenty Sixth Very Large Databases (VLDB) Conference, pages 419428, Las Vegas, 2001.
8. D. Gusfield and R.W.Irving. The Stable Marriage Problem: Structure and Algorithms. MIT Press, Cambridge, MA, 1989.
9. R. Hull. Managing semantic heterogeneity in databases: A theoretical perspective. In Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), pages 5161. ACM Press, 1997.
10. B. Korte and J. Vygen. Combinatorial Optimization: Theory and Algorithms. Springer, second edition, 2002.
11. J. Madhavan, P.A. Bernstein, and E. Rahm. Generic schema matching with Cupid. In Proceedings of the International conference on very Large Data Bases (VLDB), pages 4958, Rome, Italy, September 2001.
12. S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In Proceedings of the IEEE CS International Conference on Data Engineering, pages 117140, 2002.
13. E. Rahm and P.A. Bernstein. A survey of approaches to automatic schema matching. VLDB Journal, 10(4):334350, 2001.
14. A. Sheth and J. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys, 22(3):183236, 1990.

Reputation Ontology for Reputation Systems

Elizabeth Chang[1], Farookh Khadeer Hussain[1], and Tharam Dillon[2]

[1] Centre for Extended Enterprise and Business Intelligence
and School of Information Systems, Curtin University of Technology,
Western Australia 6845
{Elizabeth.Chang, Farookh.Hussain}@cbs.curtin.edu.au
[2] Faculty of Information Technology, University of Technology, Sydney,
Broadway, Australia
tharam@it.uts.edu.au

Abstract. The growing development of web-based reputation systems in the 21st century will have a powerful social and economic impact on both business entities and individual customers, because it makes transparent quality assessment on products and services to achieve customer assurance in the distributed web-based Reputation Systems. The web-based reputation systems will be the foundation for web intelligence in the future. Trust and Reputation help capture business intelligence through establishing customer trust relationships, learning consumer behavior, capturing market reaction on products and services, disseminating customer feedback, buyers' opinions and end-user recommendations. It also reveals dishonest services, unfair trading, biased assessment, discriminatory actions, fraudulent behaviors, and un-true advertising. The continuing development of these technologies will help in the improvement of professional business behavior, sales, reputation of sellers, providers, products and services. Given the importance of reputation in this paper, we propose ontology for reputation. In the business world we can consider the reputation of a product or the reputation of a service or the reputation of an agent. In this paper we propose ontology for these entities that can help us unravel the components and conceptualize the components of reputation of each of the entities.

1 Introduction

In this paper we propose two distinct definitions of reputation. The basic definition gives a simplistic view of reputation, and based on this simplistic view of reputation we propose ontology for reputation called the *Basic Reputation Ontology*. The basic definition and the basic reputation ontology are presented in Section 2.

The sophisticated definition of reputation gives a complete picture of reputation. We call this sophisticated definition of reputation an advanced definition of reputation, and based on this definition we define ontology for reputation termed as *Advanced Reputation Ontology*. The advanced definition and the advanced reputation ontology are presented in Section 3.

Reputation by itself is a generic term. In a service oriented or a business environment we may in fact refer to the reputation of a trusted agent, or the reputation of a product or service. Due to this we will have a specified and a specialized definition of the reputation of a product, service or trusted agent. Based on the

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 957 – 966, 2005.
© Springer-Verlag Berlin Heidelberg 2005

specialized definitions of reputation of product, reputation of service and the reputation of a trusted agent, in this paper we will propose reputation ontology for each of these business entities. The ontology for reputation of a trusted agent is presented in Section 4. The ontology for reputation of a service and the ontology for the reputation of a product are presented in Section 5 and Section 6 respectively.

In Section 7 we present ontology for the trustworthiness about an *opinion communicated by a recommender.* Finally Section 8 concludes the paper.

From existing literature we note that there has been no effort to define ontology for reputation based on the finer granularity of defining reputation (Rahman et al 2003, Aberer et al 2003, Cornelli et al 2003, Xiong et al 2003, Yu et al 2002).

2 Basic Reputation Ontology

Reputation is about developing the measure of trustworthiness from Third Party Agent's recommendations, not by the Trusting Agents themselves. This is because the Trusted Agent is unknown to the Trusting Agent.

2.1 Basic Reputation Ontology

The Reputation of a Trusted *Agent is an aggregated Reputation Value that is recommended by all of the Third Party Recommendation Agents*.

The Reputation Value is known as the Reputation of the Trusted Agent. It is an aggregated Trust Value obtained from all of the Recommendation Agents who responded to a Reputation Query.

There are several methods used to aggregate the feedback. Discussing them would be out side the scope of the paper; however, the premise in calculating the basic reputation of a Trusted Agent is outlined below:

Basic Reputation of the Trusted Agent = $\bigcup$ (Recommendation Value)

where we define $\bigcup$ as an operator for combining the Recommendation Value.

A graphical view of the Basic Reputation Ontology is shown in the following diagram though the use of UML-OCL notation.

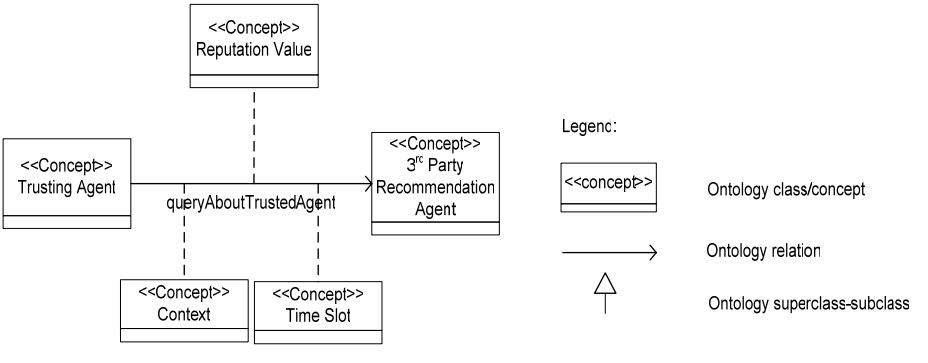


Fig. 1. Ontology for Basic Reputation of the Trusted Agent

In the above ontology diagram (Figure 1), boxes represent ontological concept, up-Arrow represent super class and sub class of concepts. Note that in ontology, there is no need to explicitly define what kind of relationship the super class (upper class) has with sub-class. The most important thing in Ontology is to build a relationship between the concepts, whether it is super-sub class hierarchy relationship or direct association (non-hierarchy). A line with an arrow represents that one concept is closely related to another. A Dotted line represents navigation to association concept. Association classes are used for associations that themselves participate in an association with another class.

Below is a formula table for the Basic Reputation Ontology:

Table 1. Formal Axiom Table of the Basic Reputation Ontology

Formula Name	Formula of Basic Reputation Value
Concept	Reputation Value
Inferred Attribute	Basic Value
Formula	Basic Value = U(Recommendation Value)
Description	Basic Reputation Value of the Trusted Agent
Variable	Recommendation Value
Ad hoc binary relation	QueryAboutTrustedAgent

With the simple (or Basic) Reputation Measure, there could be three problems created:

a) It may end up without a normal distribution in statistical analysis, such as 99% of Third Party Recommendation Agents giving 'positive' or 'trustworthy' ratings to 99% of Agents (see e-Bay example in Figure 9.8).
b) It may create doubt on the accuracy and adequacy of the Reputation Measure itself, such as the truthfulness of the Reputation Rating and the depth of the criteria addressed in the reputation.
c) It may lack addressing the dynamic nature of Trust and Reputation, as Trust and Reputation will change over time. A simple 'one value for the lifetime' is not convincing, as many assumptions may not be explored and explained clearly to the end customer and end user.

Therefore, there is a need to use a more sophisticated measurement method for Reputation. This is introduced in the next section.

3 Advanced Reputation Ontology

Advanced reputation measurement methodologies, utilize more sophisticated statistical methods to determine the reputation of a given entity. They have an impact

on the accuracy of Reputation measure, thus influencing the quality and moral hazards of service-oriented environments.

3.1 Advanced Reputation Ontology

The Reputation of a Trusted Agent is *an aggregated Reputation Value that is recommended by all of the Third Party Recommendation Agents. The aggregation is weighted by the Trustworthiness of the Recommendation Agent, the Trustworthiness of the opinion and the ranking of the 1st, 2nd and 3rd hand opinions.*

Mathematically the afore mentioned definition of reputation can be represented as:

Advanced Reputation of the Trusted Agent = $\bigcup$ (Recommendation Value * Trustworthiness of opinion * Perceived 1st. 2nd and 3rd hand opinion * Time elapsed factor)

Where we define $\bigcup$ is an operator for combining and taking into account the Trustworthiness of the Recommendation Agent's opinion, ratio of 1st hand, 2nd hand and 3rd hand opinion, and time factors. This advanced aggregation formula will enable the system to eliminate recommendations that are not trustworthy, self-recommendations, and those that are malicious.

A graphical view of the Advanced Reputation Ontology is shown in the following diagram through the use of UML-OCL notation.

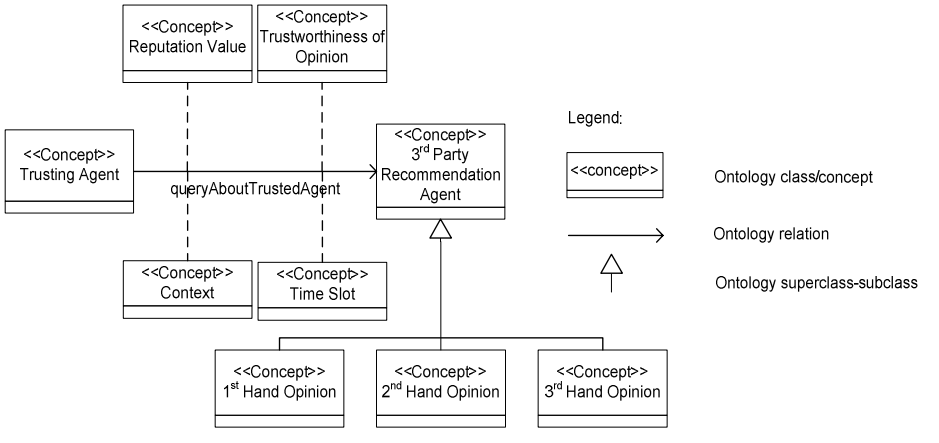


Fig. 2. Ontology for Advanced Reputation of the Trusted Agent

In the above ontology diagram (Figure 2) boxes represent ontological concept, up-Arrow represent super class and sub class of concepts, and a line with an arrow shows that one concept is closely related to another. Dotted line represents navigation to association concept. Association classes are used for associations that themselves participate in an association with another class.

Below is a formula table for the Advanced Reputation Ontology:

Table 2. Formal Axiom Table of the Advanced Reputation Ontology

Formula Name	Formula of Advanced Reputation Value
Concepts	Reputation Value, Trustworthiness of Opinion, Timeslot, 1st Hand Opinion, 2nd Hand Opinion, 3rd Hand Opinion
Inferred Attribute	Advanced Value
Formula	Advanced Value = U(Recommendation Value*Trustworthiness Value*{1st Hand Opinion Value, 2nd Hand Opinion Value, 3rd Hand Opinion Value}*Time Elapsed Factor)
Description	Advanced Reputation Value of the Trusted Agent
Variables	Recommendation Value, Trustworthiness Value, 1st Hand Opinion Value, 2nd Hand Opinion Value, 3rd Hand Opinion Value, Time Elapses Factor
Ad hoc binary relation	QueryAboutTrustedAgent

4 Ontology for Reputation of Agent

4.1 Ontology for Reputation of Agent

The Reputation of a Trusted *Agent is an aggregated reputation value that is aggregated by the recommendations from all of the Third Party Recommendation Agents. The aggregation is weighted by the Trustworthiness of the Recommendation Agent, the Trustworthiness of the opinion and the ranking of the 1st, 2nd and 3rd hand opinions that the Trusting Agent obtains through the Reputation Query about the Trusted Agent in a given context and at a given timeslot.*

4.2 Conceptual View of the Ontology for Reputation of Agent

The graphical view of the Reputation of Agent Ontology is shown in the following diagram below though the use of UML-OCL notation.

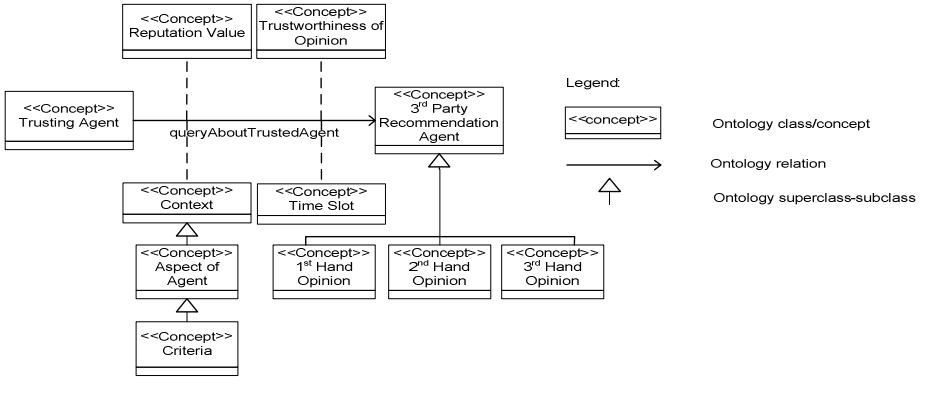


Fig. 3. Ontology for Reputation of Agent

In the above ontology diagram (Figure 3), boxes represent ontological concept, up-arrow represent super class and sub class of concepts. A line with an arrow represents that one concept is closely related to another. A Dotted line represents navigation to association concept. Association classes are used for associations that themselves participate in an association with another class.

5 Ontology for Reputation of Service

The ontology for the Reputation of Services has potential implications for the large growing number of service providers to join e-services. In this section we discuss the use of ontology for the Reputation and the Quality of Service.

5.1 Ontology for Reputation of Service

The Reputation of the quality of a Service is an aggregated reputation value that is aggregated by the recommendations from all of the Third Party Recommendation Agents. The aggregation is weighted by the Trustworthiness of the Recommendation Agent, the Trustworthiness of the opinion and the ranking of the 1st, 2nd and 3rd hand opinions that the Trusting Agent obtains though the Reputation Query about the Trusted Agen, in a given context and at a given timeslot.

5.2 Conceptual View of the Ontology for Reputation of Service

The graphical view of the Reputation of Service Ontology is shown in the following diagram though the use of UML-OCL notation.

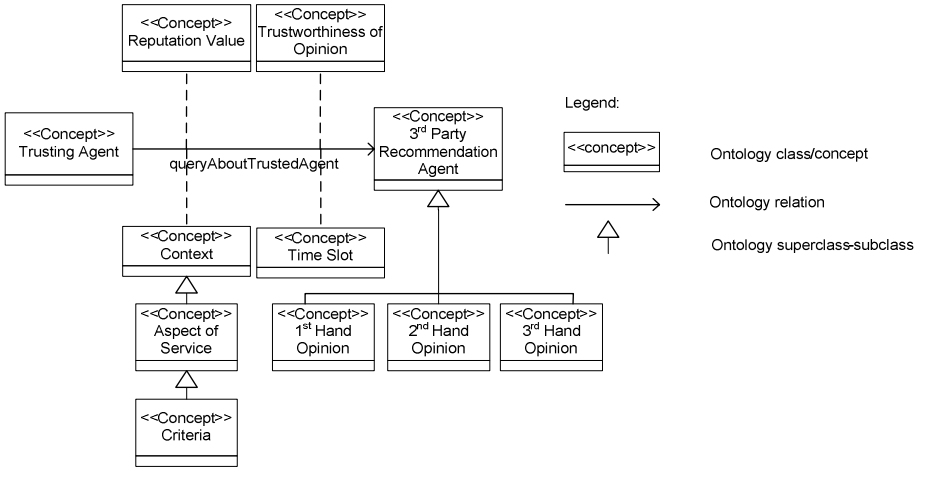


Fig. 4. Ontology for Reputation of Service

In the above ontology diagram (Figure 9.13), boxes represent ontological concept, up-Arrow represent super class and sub class of concepts. A line with an arrow represents that one concept is closely related another. A dotted line represents

navigation to association concept. Association classes are used for associations that themselves participate in an association with another class.

6.1 Ontology for Reputation of Product

The Reputation of the quality of a Product *is an aggregated reputation value that is aggregated by the recommendations from all of the Third Party Recommendation Agents. The aggregation is weighted by the Trustworthiness of the Recommendation Agent, the Trustworthiness of the opinion and the ranking of the 1st, 2nd and 3rd hand opinions that the Trusting Agent obtains through the Reputation Query about the Trusted Agen, in a given context and timeslot.*

6.2 Conceptual View of the Ontology for Reputation of Product

The graphical view of the Reputation of Product Ontology is shown in the following diagram though the use of UML-OCL notation.

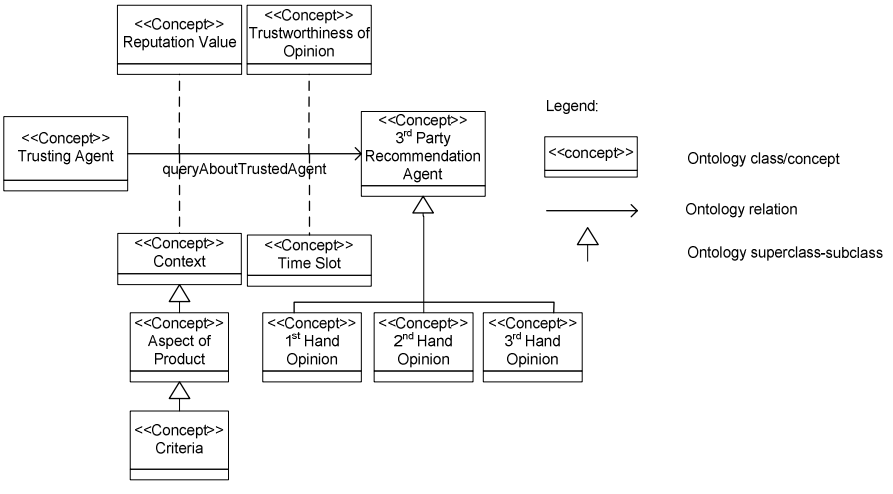


Fig. 5. Ontology for Reputation of Product

In the above ontology diagram (Figure 4.9) boxes represent ontological concept, up-Arrow represent super class and sub class of concepts. A line with an arrow represents that the concept is closely related to another concept. Dotted line represents navigation to association concept. Association classes are used for associations that themselves participate in an association with another class.

7 Trustworthiness of Opinion Ontology

7.1 Opinions in Reputation

The most crucial factor for reputation measurement (of a trusted agent or a service or a product) is the validation of trustworthiness of the opinion or the recommendation

provided by the Third Party Recommendation Agents. The trusting entity after soliciting recommendation from the third party recommendation Agents needs to have an idea of the extent to which it regards each of the recommendations communicated by each of the third party recommendation Agents as being correct. In other words it needs to make known the trustworthiness of the opinion communicated by the third party agent so that the communicated recommendation can be properly weighted. Discussing the mathematical framework for determining the trustworthiness of the opinion is outside the scope of this paper. Further discussion along with detailed examples of how to determine the trustworthiness of the opinion can be found in (Chang, Dillon and Hussain, 2005). In this paper we will provide a ontology for the trustworthiness of the opinion.

7.2 Ontology for Trustworthiness of Opinion

We define the Opinion Trust Ontology as the following Trust Tuple:

Review Trust [Receiver, Reviewer, Review or Feedback, Assessment Criteria, Timeslot, and Trustworthiness of each assessment criterion)

The graphical view of the Trustworthiness of Opinion Ontology is shown in the following diagram though the use of UML-OCL notation.

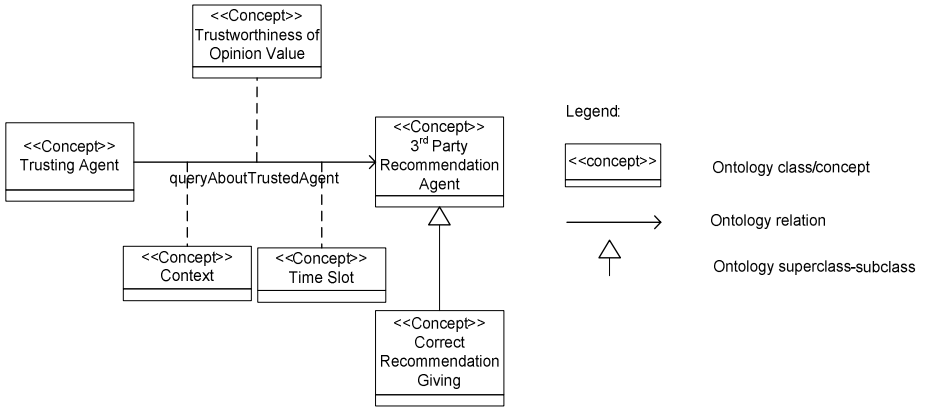


Fig. 7. Ontology for the Trustworthiness of Opinion

In the above ontology diagram (Figure 7), boxes represent ontological concept, up-Arrow represent super class and sub class of concepts. A line with arrow represents that the concept is closely related to another concept. Dotted line represents navigation to association concept. Association classes are used for associations that themselves participate in an association with another class.

Below is a table for the Trustworthiness of Opinion Ontology:

Table 4. Formal Axiom Table of the Trustworthiness of Opinion Ontology

Formula Name	Formula of Trustworthiness of Opinion Value
Concepts	Trustworthiness of Opinion Value, Correct Recommendation Giving
Inferred Attribute	Trustworthiness Value
Formula	Trustworthiness Value = Actual Trust Value Found On Interaction - Recommendation Value
Description	Trust Value of Recommendation Agent in giving correct opinion
Variables	Actual Trust Value Found On Interaction, Recommendation Value
Ad hoc binary relation	QueryAboutTrustedAgent

8 Application of Reputation Ontology and Technology

Reputation Systems address the quality of goods and services, sellers or service providers, network agents or reviewers, which is based on a number of criteria. Currently, some well known e-commerce portals already start using the reputation systems, such as BizRate, Slashdot, Elance, BBC, Alibris, MoneyControl, Yahoo, Epinions, eBay and CNET. Other popular websites have also adopted reputation systems such as KuroHin.org, Reel.com, Amazon, CDNow.com, GroupLens and MovieLens and CitySearch, to name a few. However, the Reputation technology only adopted at a high level, such as only ranking the products and fewer ranking basic services, and the reputation value is aggregated through simple formulae.

	BizRate	eLance	Alibris	Money Control	Yahoo	Epinions	eBay	CNET	Movie Lens
Rating Merchants Reputation	●	●	●		●	●	●	●	
Rating Products Reputation	●		●		●	●		●	●
Rating Customers Reputation		●					●		
Rating Reviews Reputation				●					
Rating Reviewers Reputation				●					
Recommendation Systems									●

The above table gives a high-level view of the technology adoption (the black-dots) of the listed companies (see horizontal bar) for their business intelligence. Due to the space constraints of this paper, we will not introduce their site; however, readers are encouraged to visit website themselves.

9 Conclusions

In this paper we propose a basic and an advanced definition of reputation, and based on this definition we proposed the basic reputation ontology and the advanced reputation ontology respectively. Additionally, we proposed specialized reputation ontology for the reputation of a product, the reputation of a service or the reputation of the trusted agent. Finally we proposed ontology for the trustworthiness of the opinion of a recommendation.

References

Chang, E. Dillon T, Hussain, F "Trust and Reputation for Service Oriented Environment". John Wiley and Sons, 2005

Abdul-Rahman, A., & Hailes, S., (2003), *Relying On Trust To Find Reliable Information*, Available: http://www.cs.ucl.ac.uk/staff/F.AbdulRahman/docs/dwacos99.pdf (7/08/2003).

Aberer, K. & Despotovic, Z., (2003), *Managing Trust in a Peer-2-Peer Information System*, Available: http://citeseer.nj.nec.com/aberer01managing.html (11/9/2003).

Ba, S., and Pavlou, P., (2002). "Evidence of the Effect of Trust Building Technology in Electronic Markets: Price Premiums and Buyer Behavior". *MIS Quarterly*, 26 (3).

Bakos, Y., and Dellarocas, C., (2002) "Cooperation without Enforcement?-A comparative Analysis of Litigation and Online Reputation as Quality Assurance Mechanism". *Proceedings of the 23rd International Conference on Information Systems* (ICIS 2002), Barcelona, Spain.

Cornelli, F., Damiani, E., Vimercati, S., De Capitani di Vimercati, Paraboschi, S. & Samarati, P., (2003), *Choosing Reputable Servents in a P2P Network*, Available: http://citeseer.nj.nec.com/cache/papers/cs/26951/http:zSzzSzseclab.crema.unimi.itzSzPaperszSzwww02.pdf/choosing-reputable-servents-in.pdf (20/9/2003).

Dellarocas C., (2003) "The Digitization of Word-of-mouth: Promise and Challenges of Online Feedback Mechanisms" *Working Paper*, March 2003, Massachusetts Institute of Technology, http://ssrm.com

Xiong, L. & Liu, L., (2003), A *Reputation-Based Trust Model for Peer-to-Peer eCommerce Communities*, Available: http://citeseer.nj.nec.com/xiong03reputationbased.html (9/10/2003).

Yu, B., Singh, M. P., (2002) . 'Distributed Reputation Management for Electronic Commerce.' Computation Intelligence 18 (4) : 535-549.

Translating XML Web Data into Ontologies

Yuan An and John Mylopoulos

University of Toronto, Canada
{yuana, jm}@cs.toronto.edu

Abstract. Translating XML data into ontologies is the problem of finding an instance of an ontology, given an XML document and a specification of the relationship between the XML schema and the ontology. Previous study [8] has investigated the *ad hoc* approach used in XML data integration. In this paper, we consider to translate an XML web document to an instance of an OWL-DL ontology in the Semantic Web. We use the semantic mapping discovered by our prototype tool [1] for the relationship between the XML schema and the ontology. Particularly, we define the *solution* of the translation problem and develop an algorithm for computing a *canonical solution* which enables the ontology to answer queries by using data in the XML document.

1 Introduction

XML has become an accepted standard for publishing data on the Web. To integrate XML data, a former paper [8] has studied the *ad hoc* approach to translating various XML documents into a central ontology instance. In this paper, we study a generic and a formal framework for translating an XML document into an instance of an ontology. The following example illustrates the problem. Suppose we have an XML document $\mathcal{X}$:

```
<db>
   <student sname=''Jerry''>
      <takes>
         <course title=''Database Theory''/>
         <course title=''Combinatorial Optimization''/>
      </takes>
      <advisor pname=''John''/>
   </student>
</db>
```

Suppose we have an ontology shown graphically in Figure 1 using UML notation. Given a natural mapping semantically relating the XML schema to the ontology, we would expect that an instance of the ontology contains the following assertions: *Student*(t_1), *Course*(t_2), *Course*(t_3), *Professor*(t_4), *hasName*(t_1,"Jerry"), *hasTitle*(t_2,"Data base Theory"), *hasTitle*(t_3,"Combinatorial Optimization"), *hasName* (t_4,"John"), *takes*(t_1,t_2), *takes* (t_1,t_3), *hasAdvisor*(t_1,t_4), *Professor*(u_1), *Professor*(u_2), *Course*(u_3), *teaches*(u_1,t_2), *teaches*(u_2, t_3), *teaches*(t_4,u_3), where t_i $i = 1, ..., 4$ and u_j $j = 1, ...3$ are anonymous individuals in the ontology.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 967–976, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Note that the difference between t_is and u_js is that the original XML document provides no information about the individuals u_1, u_2 and u_3. They were deduced by the ontology constraints. However, if we replace u_1 and u_2 by t_4, then the resulting instance will still satisfy all constraints and it says that professor t_4 teaches both courses t_2 and t_3. Alternatively, we could also construct an instance in which the professor t_4 teaches only the course t_2, while the course t_3 is taught by some unknown professor u_2. This tells us that there could be different instances that are consistent with the ontology and satisfy a given mapping from the XML schema to the ontology. So if we are given a source document $\mathcal{X}$ shown above and a query over the ontology, how can we answer it? If our query is, for example, *What is the name of the person who is the advisor of the person whose name is Jerry?* The answer is *John* regardless of a particular instance that was created for the ontology. As another example, consider the query *What is the title of the course taught by Jerry's advisor?* This query cannot be answered with certainty in this scenario.

Ontologies play a central role in the Semantic web. Recently, W3C has recommended the OWL web ontology language for describing ontologies in the Semantic Web. If an XML document needs to be translated into an OWL ontology, the resulting ontology should preserve the information in the XML document and be able to answer queries by using these information. Consequently, a translation involves specifying a mapping, checking the consistency, and preserving information. In this paper, we consider the OWL-DL ontology language because of its close relationship with Description Logics. As a result, the OWL-DL ontology language enable us to develop a translation algorithm. The overall framework is generic in the sense that the theoretical issues apply to many translation problems between databases and ontologies.

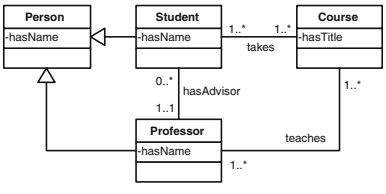


Fig. 1. An Ontology

The rest of the paper is organized as follows. Section 2 presents the formal specifications about OWL-DL ontology and XML. Section 3 defines the problem and Section 4 defines the canonical solution. Section 5 develops the algorithm, and finally, Section 6 gives the conclusions.

2 Preliminaries

We assume readers are familiar with the standard notations and semantics of Description Logics, though we summarize here one flavor relating to the OWL-DL web ontology language. OWL-DL is closely related to the $\mathcal{SHOIN}(\mathbf{D})$ description logic [5], and the meanings of its terminology can be found in [4,5].

A datatype theory $\mathbf{D}$ is a mapping from a set of datatypes to a set of values. The datatype (or concrete) domain, written $\Delta^{\mathcal{I}}_{\mathbf{D}}$, is the union of the mappings

of the datatypes. Let $\mathbf{R}$ be set of role names consisting of a set of *abstract role names* $\mathbf{R}_A$ and *a set of concrete role names* $\mathbf{R}_D$. The set of $\mathcal{SHOIN}$-roles (or roles for short) consist of a set of *abstract roles* $\mathbf{R}_A \cup \{R^- | R \in \mathbf{R}_A\}$ and a set of *concrete roles* $\mathbf{R}_D$. An *RBox* $\mathcal{R}$ consists of a finite set of transitivity axioms $\mathsf{Trans}(R)$, and role inclusion axioms of the form $R \sqsubseteq S$ and $T \sqsubseteq U$, where R and S are abstract roles, and T and U are concrete roles. $\sqsubseteq^*$ denotes the reflexive-transitive closure of $\sqsubseteq$ on roles, i.e., for two abstract roles R, S, $S \sqsubseteq^* R \in \mathcal{R}$ if S and R are the same, $S \sqsubseteq R \in \mathcal{R}$, $\mathsf{Inv}(S) \sqsubseteq \mathsf{Inv}(R) \in \mathcal{R}$, or there exists some role Q such that $S \sqsubseteq^* Q \in \mathcal{R}$ and $Q \sqsubseteq^* R \in \mathcal{R}$. A role not having transitive sub-roles is called a *simple* role, and $\mathsf{Inv}(R) = R^-$.

The set of $\mathcal{SHOIN}(\mathbf{D})$ concepts is defined by the following syntactic rules, where C_is are concepts, A is an atomic concept, R is an abstract role, S is an abstract *simple role*, T is a concrete role, o_i are individuals, D is a datatype, and n is a non-negative integer:

$$C \rightarrow A | \neg C | C_1 \sqcap C_2 | C_1 \sqcup C_2 | \exists R.C | \forall R.C | \geq nS | \leq nS | \\ \{o_1, .., o_n\} | \geq nT | \leq nT | \exists T.D | \forall T.D$$

A *TBox* $\mathcal{T}$ consists of a finite set of concept inclusion axioms $C_1 \sqsubseteq C_2$; an *ABox* $\mathcal{A}$ consists of a finite set of concept and role assertions and individual (in)equalities $C(a)$, $R(a, b)$, $a = b$, $a \neq b$, respectively. A $\mathcal{SHOIN}(\mathbf{D})$ knowledge base (an ontology) $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ consists of a TBox $\mathcal{T}$, an RBox $\mathcal{R}$, and an ABox $\mathcal{A}$. The semantics of $\mathcal{SHOIN}(\mathbf{D})$ is given by means of an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consisting of an non-empty domain $\Delta^{\mathcal{I}}$, disjoint from the datatype domain $\Delta^{\mathcal{I}}_{\mathbf{D}}$, and a mapping $\cdot^{\mathcal{I}}$, which interprets atomic and complex concepts, roles, axioms, and assertions in the standard description logic way. An interpretation $\mathcal{I}$ is a *model* of the knowledge base $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ if $\mathcal{I}$ satisfies every concept, axiom, and assertion in $\mathcal{O}$. From the database perspective, the TBox $\mathcal{T}$ and the RBox $\mathcal{R}$ can be viewed as a schema with unary and binary relational tables, and the ABox $\mathcal{A}$ can be viewed as an instance. An ABox $\mathcal{A}$ is *consistent* with respect to $\mathcal{O}$ if there is a model of $(\mathcal{T}, \mathcal{R}, \mathcal{A})$ (we say $\mathcal{O}$ is consistent). A concept or role assertion β is a *logical consequence* of an ABox $\mathcal{A}$ (written $\mathcal{A} \models \beta$), if for every model of $\mathcal{A}$ w.r.t $< \mathcal{T}, \mathcal{R} >$, β is true. We write $\mathcal{O} \models \beta$ for β is a *logical consequence* of the ontology.

An XML document is typically modeled as a node-labeled tree. For our purpose, we assume that each XML document is described by an XML schema consisting of a set of element and attribute type definitions. Specifically, we assume the following countably infinite disjoint sets: **Ele** of element names, **Att** of attribute names, and **Dom** of simple type names including the built-in XML schema datatypes. Attribute names are preceded by a "@" to distinguish them from element names. Given finite sets $E \subset$**Ele** and $A \subset$**Att**, a XML schema $\mathcal{S} = (E, A, \tau, \rho, \kappa)$ specifies the type of each element ℓ in E, the attributes that ℓ has, and the datatype of each attribute in A. Specifically, An element type τ is defined by the grammar $\tau ::= \epsilon | \mathsf{Sequence}[\ell_1 : \tau_1, ...\ell_n : \tau_n] | \mathsf{Choice}[\ell_1 : \tau_1, .., \ell_n : \tau_n]$ $(\ell_1, .., \ell_n \in E)$, where ϵ is for the empty type, and Sequence and Choice are complex types. Each element associates an occurrence constraint with two values: *minOccurs* indicating the minimum occurrence and *maxOccurs* indicating the

maximum occurrence. The set of attributes of an element $\ell \in E$ is defined by the function $\rho : E \rightarrow 2^A$; and the function $\kappa : A \rightarrow$**Dom** specifies the datatypes of attributes in A. Each datatype name associates with a set of values in a domain *Dom*. In this paper, we do not consider the *simple type elements* (corresponding to DTD's **PCDATA**), assuming instead that they have been represented using attributes. Furthermore, a special element $\underline{r} \in E$ is the root of the XML schema such that $\rho(\underline{r}) = \emptyset$, and we assume that for any two element $\ell_i, \ell_j \in E$, $\rho(\ell_i) \cap \rho(\ell_j) = \emptyset$.

An XML document $\mathcal{X} = (N, <, \underline{r}, \lambda, \eta)$ over (E, A) consists of a set of nodes N, a child relation $<$ between nodes, a root node $\underline{r}$, and two functions such as:

- a labeling function λ:$N \rightarrow E \cup A$ such that if $\lambda(v) = \ell \in E$, we say that v is in the element type ℓ; if $\lambda(v) = @a \in A$, we say that v is an attribute $@a$;
- a partial function η:$N \rightarrow$ *Dom* for every node v with $\lambda(v) = @a \in A$, assigning values in domain *Dom* that supplies values to simple type names in **Dom**.

An XML document $\mathcal{X} = (N, <, \underline{r}, \lambda, \eta)$ conforms to a schema $\mathcal{S} = (E, A, \tau, \rho, \kappa)$, denoted by $\mathcal{X} \models \mathcal{S}$, if:

1. for every node v in $\mathcal{X}$ with children $v_1, .., v_m$ such that $\lambda(v_i) \in E$ for $i = 1, ..., m$, if $\lambda(v) = \ell$, then $\lambda(v_1)$,..., $\lambda(v_m)$ satisfies $\tau(\ell)$ and the occurrence constraints.
2. for ever node v in $\mathcal{X}$ with children $u_1, ..., u_n$ such that $\lambda(u_i) = @a_i \in A$ for $i = 1, ..., n$, if $\lambda(v) = \ell$, then $\lambda(u_i) = @a_i \in \rho(\ell)$, and $\eta(u_i)$ is a value having datatype $\kappa(@a_i)$.

Now we turn to the mapping language relating a pattern in an XML schema with a formula in an ontology. On the XML side, the basic component is *attribute formulas* [2], which are specified by the syntax $\alpha ::= \ell | \ell(@a_1 = x_1, .., @a_n = x_n)$, where $\ell \in E$, $@a_1, .., @a_n \in A$, E and A are element names and attribute names respectively; and variables $x_1, .., x_n$ are the free variables of α. Tree-pattern formulas over an XML schema $\mathcal{S} = (E, A, \tau, \rho, \kappa)$ are defined by $\psi ::= \alpha | \alpha[\varphi_1, .., \varphi_n]$, where α ranges over attribute formulas over (E, A). The free variables of a tree formula ψ are the free variables in all the attribute formulas that occur in it. For example, $Company[Department[employee(@eid = x_1)[manager\ (@mid = x_2)\ [employee(\ @eid = x_3)\ ]]]]$ is a tree formula.

An attribute formula is evaluated in a node of an XML document, and values for free variables come from domain *Dom*. If $\mathcal{X}$ is an XML document over (E, A) and v a node of $\mathcal{X}$, then

- $(\mathcal{X}, v) \models \ell$ iff $\lambda(v) = \ell$, for $\ell \in E$.
- if $\alpha(x_1, ..., x_n) = \ell(@a_1 = x_1, .., @a_n = x_n)$, then $(\mathcal{X}, v) \models \alpha(s_1, ..., s_n)$, where $s_1, ..., s_n \in Dom$, iff $\lambda(v) = \ell$, and for each child v_i of v such that $\lambda(v_i) = @a_i$, $\eta(v_i) = s_i$ for $i \in [1, ..n]$.

Given a document $\mathcal{X}$, a tree-pattern formula $\psi(\overline{x})$, and a tuple $\overline{s}$ from *Dom*, $\psi(\overline{s})$ is satisfied in $\mathcal{X}$ (written $\mathcal{X} \models \psi(\overline{s})$) if there is a *witness* node v for $\psi(\overline{s})$. Formally, a *witness* node for a $\psi(\overline{s})$ is defined as follows:

- v is a witness node for $\alpha(\overline{s})$, where α is an attribute formula, iff $(\mathcal{X}, v) \models \alpha(\overline{s})$.
- v is witness node for $\alpha(\overline{s})[\ \psi_1(\overline{s}_1),...,\psi_m(\overline{s}_m)]$ iff $(\mathcal{X}, v) \models \alpha(\overline{s})$ and there are m children $v_1, ..., v_m$ of v such that each v_i is a witness node for $\psi_i(\overline{s}_i)$, for $i = 1, ..., m$.

On the ontology side, we use conjunctive formulas with annotations, which treat atomic concepts and roles as unary and binary predicates, respectively. For example, given an ontology containing the atomic concept *Employee* and roles *hasId*, *hasManager*, and *manages*, the following is a mapping formula,

$$
\begin{array}{l}
Company[Department[\\
\quad employee(@eid = x_1)[\\
\quad\quad manager\ (@mid = x_2)\ [\\
\quad\quad\quad employee\ (@eid{=}x_3)\ ]]]] \rightarrow \\
Employee(Y_1),\ hasId(Y_1, x_1),\ Employee(Y_2),\ hasId(Y_2, x_2), \\
hasManager(Y_1, x_2),\ Employee(Y_3),\ hasId(Y_3, x_3),\ manages(Y_2, Y_3):: \\
identif(Y_1, x_1),\ identif(Y_2, x_2),\ identif(Y_3, x_3).
\end{array}
$$

There are two sorts of variables. One sort of variables denoted, e.g., by Y_is, represent the individuals in the ontology, and another sort of variables denoted, e.g., by x_js, represent data values containing the attribute values in the XML document and concrete values in the ontology. Since attribute values in the XML document come from the domain *Dom*, while concrete values in the ontology come from domain $\Delta^{\mathcal{I}}_{\mathbf{D}}$, we assume that each mapping formula implies a set of conversion functions such that when the single variable name x_j is used on both sides, both datatypes in the corresponding positions are matched through an implicit conversion function. We denote by ConstValue the set of all data values that occur in the XML document and we also call them *constant values.* In addition, we assume an infinite set VarValue which we call *variable values* including an infinite set Individual of *individuals* and an infinite set DataValue of *data values.* We require that ConstValue$\cap$VarValue$=\emptyset$.

The annotation comes after :: in the mapping formula. Each predicate in the annotation is of the form $identif(Y, \overline{Z})$ in which Y is an individual variable and $\overline{Z}$ is a tuple of variables. The meaning of $identif(Y, \overline{Z})$ is as follows. The information in XML document indicates that an individual belonging to the concept C in which Y is the placeholder variable, i.e, $C(Y)$ appearing in the formula, can be identified by a set of roles P_1,...,P_n in the ontology, whereas P_1,...,P_n bind Y with $\overline{Z}$ in the formula, i.e., $P_1(Y, Z_1)$,...,$P_n(Y, Z_n)$ appear. We will see later that the annotation is important during the translation and for consistency checking in the ontology. To specify the mapping formulas, we have proposed a semi-automatic tool MAPONTO in [1].

3 The Problem of Translating XML Data into Ontologies

We now define the problem of translating XML into ontologies (X-to-O).

Definition 1 (Semantics of Mapping Formulas). *Given an XML schema $\mathcal{S}$ and an ontology $\mathcal{O} =< \mathcal{T}, \mathcal{R}, \mathcal{A} >$, a mapping formula is an expression of the form:*

$$\Psi : \psi(\overline{x}) \rightarrow \varphi(\overline{Y}, \overline{x}) :: annotation. \quad (1)$$

where $\psi(\overline{x})$ is a tree-pattern formula over $\mathcal{S}$, $\varphi(\overline{Y}, \overline{x})$ is a conjunctive formula over atomic concept and role names of $\mathcal{O}$, and $\overline{Y}$ and $\overline{x}$ have no variables in common.

Given an XML document $\mathcal{X}$ conforming to $\mathcal{S}$ and an ontology instance $\mathcal{A}$ consistent with $\mathcal{O}$, we say that the pair $< \mathcal{X}, \mathcal{A} >$ satisfies the formula (1) if whenever there is a tuple $\overline{s}$ such that $\mathcal{X} \models \psi(\overline{s})$, there exists a tuple $\overline{t}$ such that for each assertion β in the formula $\varphi(\overline{t}, \overline{s})$, $\mathcal{A} \models \beta$.

Definition 2 (X-to-O problem). *The problem of translating XML data into ontologies (X-to-O) is a triple $(\mathcal{S}, \mathcal{O}, \Sigma_{\mathcal{SO}})$, where $\mathcal{S} =< E, A, \tau, \rho, \kappa >$ is an XML schema, $\mathcal{O} =< \mathcal{T}, \mathcal{R}, \mathcal{A} >$ is an ontology, and $\Sigma_{\mathcal{SO}}$ is a set of mapping formulas between $\mathcal{S}$ and $\mathcal{O}$.*

Definition 3 (Solutions). *Given an X-to-O problem $\mathcal{P} = (\mathcal{S}, \mathcal{O}, \Sigma_{\mathcal{SO}})$ and an XML document $\mathcal{X}$ conforming to $\mathcal{S}$, an instance $\mathcal{A}$ consistent with $\mathcal{O}$ such that $< \mathcal{X}, \mathcal{A} >$ satisfies all formulas in $\Sigma_{\mathcal{SO}}$ is called a solution for $\mathcal{P}$.*

Recall the *Data Exchange* problem [3,2]. A data exchange setting is a tuple $(\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$, where $\mathbf{S}$ is a source schema, $\mathbf{T}$ is a target schema, Σ_{st} is a set of *source-to-target dependencies,* or *STDs*, that express the relationship between $\mathbf{S}$ and $\mathbf{T}$, and Σ_t is a set of constraints on the target schema. A solution of the data exchange problem is an instance J over the target schema $\mathbf{T}$ when given an instance I over the source schema $\mathbf{S}$, such that I and J together satisfy all formulas in Σ_{st} and Σ_t. In general, there may be many different solutions for a given instance I, and under target constraints, there may be no solutions at all. If one poses query Q over the target schema, and a source instance I is known, the usual semantics in data exchange uses *certain answers*. A key problem in data exchange is to find a particular solution J_0 so that $\underline{certain}(Q, I)$ can be obtained by evaluating some query over J_0.

Coming back to X-to-O problem, we have defined that the source is an XML schema and the target is an ontology. By analogy, our mapping formulas are the source-to-target dependencies that express the relationships between the XML schema and the ontology. Given an XML document conforming to the XML schema, we want to compute a consistent instance of the ontology, such that the XML document together with the ontology instance satisfy all formulas in the mapping. The major difference from the data exchange problem is that with an ontology as the target, computing a solution calls for a different algorithm.

4 Canonical Solution

As illustrated by the example in Section 1, there could be many solutions for an X-to-O problem. In this section, we define the *canonical solution* in terms of answering queries against ontologies. For the query language, we use a simple conjunctive query language (CQ_0) which can represent most of the proposed

query languages for RDF data (e.g., SPARQL [7]). Formally, a CQ_0 query is of the form:

$$Q : q(\overline{x}) \leftarrow p_1(\overline{Y_1}), p_2(\overline{Y_2}), ..., p_n(\overline{Y_n}). \tag{2}$$

where $\overline{x}$ is a tuple of variables or constants which take values from concrete (datatype) domains (e.g., Integer, String, etc.), $\overline{Y_1}, ..., \overline{Y_n}$ are tuples of variables or constants which take values from both concrete domains and individuals (e.g., object identifiers), and we require $\overline{x} \subset \overline{Y_1} \cup ... \cup \overline{Y_n}$. The predicates $p_1, ..., p_n$ are atomic concept and (abstract and concrete) role names in an ontology. The predicate q is an ordinary predicate with an arity $m = |\overline{x}|$. Let $\underline{\text{Const}}$ denote the set of constants appearing in an ontology, $\underline{\text{Var}}$ denote a set of variables, and $\underline{\text{Const}} \cap \underline{\text{Var}} = \emptyset$. Let $h : \underline{\text{Const}} \cup \underline{\text{Var}} \rightarrow \underline{\text{Const}}$ be a mapping from a tuple of variables or constants to a tuple of constants such that if $c \in \underline{\text{Const}}$, $h(c) = c$. Given an ontology $\mathcal{O} =< \mathcal{T}, \mathcal{R}, \mathcal{A} >$ with an instance $\mathcal{A}$, the answer of the query (2) is defined as a set of tuples of concrete (datatype) values $\{\overline{s}\}$ such that for each tuple $\overline{s}$ there is a mapping h such that $h(\overline{x}) = \overline{s}$ and there are tuples $h(\overline{Y_i})$, $\mathcal{O} \models p_i(h(\overline{Y_i}))$ for each $i = 1, ..., n$. A CQ_0 query only returns tuples consisting of datatype values.

Assume that we are given an X-to-O problem $(\mathcal{S}, \mathcal{O}, \Sigma_{\mathcal{SO}})$, an XML document $\mathcal{X}$ conforming to $\mathcal{S}$, and a CQ_0 query Q against the ontology. What does it mean to answer Q? As in the data exchange problem [3], since there may be many possible solutions to the X-to-O problem, we define the semantics of Q in terms of *certain answers*:

$$\underline{certain}(Q, \mathcal{X}) = \bigcap_{\mathcal{A}' \text{ is a solution}} Q(\mathcal{A}') \tag{3}$$

where, $Q(\mathcal{A}')$ is the answers of the query Q evaluated over the solution $\mathcal{A}'$. Thus, a tuple $\overline{s}$ of datatype values is in the set of *certain answers* $\underline{certain}(Q, \mathcal{X})$, if $\overline{s} \in Q(\mathcal{A}')$ for every solution $\mathcal{A}'$ of the X-to-O problem.

Definition 4 (Canonical Solution). *Given an X-to-O problem and a CQ_0 query Q against the ontology. A solution $\mathcal{A}$ is a canonical solution if it produces the certain answers when given an XML document $\mathcal{X}$ conforming to the XML schema.*

5 Computing a Canonical Solution

Given an X-to-O problem $\mathcal{P} = (\mathcal{S}, \mathcal{O}, \Sigma_{\mathcal{SO}})$, we assume that the ontology $\mathcal{O} =< \mathcal{T}, \mathcal{R}, \mathcal{A} >$ is satisfiable. For each mapping formula $\Psi : \psi(\overline{x}) \rightarrow \varphi(\overline{Y}, \overline{x}) :: annotation$, the formula $\varphi(\overline{Y}, \overline{x})$ has the form $C_i(Y_i)$,...,P_i $(Y_i,\ Y_j)$,..., $T_i(Y_i, x_{i_j})$,... where C_i is an atomic concept name, P_i is an abstract role name, T_i is a concrete role name, and $\overline{x} = \{...x_{i_j}...\}$. We assume that $\varphi(\overline{Y}, \overline{x})$ is consistent with the ontology $\mathcal{O}$ (written $\mathcal{O} \models \varphi(\overline{Y}, \overline{x})$), which means that for each model $\mathcal{I} =< \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} >$ of $\mathcal{O}$, the interpretation function $\cdot^{\mathcal{I}}$ can be extended to the variables in $\varphi(\overline{Y}, \overline{x})$ in such a way that $\mathcal{I}$ satisfies every atom in $\varphi(\overline{Y}^{\mathcal{I}}, \overline{x}^{\mathcal{I}})$.

Informally, to compute a *canonical solution* when given a mapping formula Ψ and an XML document $\mathcal{X}$, we start from an initial ontology $\mathcal{O}_0 = < \mathcal{T}, \mathcal{R}, \mathcal{A}_0 >$ and add new assertions $C_i(t_i)$, $P_i(t_i, t_j)$, and $T_i(t_i, s_{i_j})$ in turn to generate a series of ABoxes $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$..., whenever there is a tuple $\overline{s} \in \underline{\text{ConstValue}}$ such that $\mathcal{X} \models \psi(\overline{s})$. The assertions $C_i(t_i)$, $P_i(t_i, t_j)$, and $T_i(t_i, s_{i_j})$ are instantiated from the mapping formula by substituting $\overline{s}$ for $\overline{x}$ and by substituting $\overline{t}$ for $\overline{Y}$, where $\overline{t}$ is a tuple of values in $\underline{\text{Individual}}$. When adding these assertions, some extra assertions will probably be added according to axioms in TBox and RBox. There will be a finite number of ABoxes $\mathcal{A}_1$, $\mathcal{A}_2$, ...,$\mathcal{A}_n$ because there are finite number of tuples in $\mathcal{X}$ satisfying the mapping formula and the propagation we will use terminates. Before presenting the algorithm, we first describe how to generate $\overline{t}$.

Let $\underline{\text{Skolem}}$ be a set of function symbols each of which has an arity. For a function symbol f_C of arity n w.r.t. the concept C, the value of applying f_C to a set of values $d_1, ..., d_n \in \underline{\text{ConstValue}} \cup \underline{\text{Individual}}$ is denoted as $f_C(d_1, .., d_n)$. We require that $f_C(d_1, .., d_n)$ is in the set $\underline{\text{Individual}}$ and two $f_C(d_1, ..., d_n)$ are equal iff they are syntactically equivalent. We choose $\overline{t}$ as follows. Suppose the annotation of the formula Ψ has a predicate $identif(Y_i, \overline{Z})$, where $C_i(Y_i)$ is in Ψ. If $\overline{Z}$ consists of only variables in $\overline{x}$, then let $t_i = f_{C_i}([\overline{Z}/\overline{s}])$ ($[\overline{Z}/\overline{s}]$ means substituting $\overline{s}$ for the variables in $\overline{Z}$ w.r.t. the substitution of $\overline{x}$ in $\varphi(\overline{Y}, \overline{x})$); else $t_i = f_{C_i}([\overline{Z}/\overline{s} \cup \overline{t_j}])$, where each t_j is an individual chosen recursively for individual variables in $\overline{Z}$. The process terminates due to the propagation of the tree structures in XML documents in the annotation.

To detect any inconsistent ABox during the process of computing the canonical solution, we add extra assertions in addition to the assertions instantiated from the mapping formula. A set of propagation rules serves this purpose. The propagation rules are derived from the axioms in the TBox and RBox, and they only apply to the individuals constructed by Skolem functions. We assume that all inclusion axioms in TBox are *concept definition* and the TBox is *acyclic*. That means only axioms of the form $CN \sqsubseteq C$ or $CN \doteq C$ are in TBox, where CN is a concept name, and C does not directly or indirectly refer to CN.

Here are the propagation rules. For an ABox $\mathcal{A}_i$ and individuals t of the form $f_C(\overline{s})$ where f_C is a Skolem function symbol, we add new assertions which do not exist previously to $\mathcal{A}_i$ by the following rules:

1. adding $C(t)$ if $CN(t)$ and $CN \sqsubseteq C$ or $CN \doteq C$ are in TBox;
2. adding $C_1(t)$ and $C_2(t)$ if $(C_1 \sqcap C_2)(t)$ is in $\mathcal{A}_i$;
3. adding $C(t_1)$ ($v \in d$) if $(\forall R.C)(t)$ and $R(t, t_1)$ (resp. $(\forall T.d)(t)$ and $T(t, v)$) are in $\mathcal{A}_i$;
4. adding $R(t, u)$ and $C(u)$ ($T(t, v)$ and $v \in d$) if $(\exists R.C)(t)$ (resp. $(\exists T.d)(t)$) is in $\mathcal{A}_i$ and there is no $R(t, u)$ (resp. $T(t, v)$);
5. adding $P(t, t_1)$ (or $P(t_1, t)$) if $R(t, t_1)$ (or $R(t_1, t)$) is in $\mathcal{A}_i$ and $R \sqsubseteq^* P$ in RBox;
6. replacing t with one of $\{o_1, ..., o_n\}$ if $\{o_1, ..., o_n\}(t)$ is in $\mathcal{A}_i$;

7. adding $C_1(t)$ or $C_2(t)$ if $(C_1 \sqcup C_2)(t)$ is in $\mathcal{A}_i$;
8. replacing the occurrences of t_i with t_j for an individual t_i not computed by a Skolem function and an individual t_j of the form $f_C(\overline{s})$, if $(\leq nR)(t)$ is in $\mathcal{A}_i$ and there exists t_k $k = 1, ..., n+1$ such that $R(t, t_k)$ exists.

Rules 1-5 above are deterministic and rules 6-8 are nondeterministic. There is only one generating rule 4.) which generates u from VarValue but u does not use Skolem functions; therefore, the propagation will terminate for a *tree* structure with depth at most 1.

A *number restriction clash* is the situation in that some abstract role R (resp. concrete role T), $(\leq nR)(t)$ (resp. $(\leq nT)(t)$) is in $\mathcal{A}_i$ and there are $n+1$ different individuals $t_1, ..., t_{n+1}$ (resp. values $v_1, ..., v_{n+1}$) such that $R(t, t_j)$ (resp. $T(t, v_j)$) in $\mathcal{A}_i$ for $j = 1, ..., n+1$. If an ABox $\mathcal{A}_i$ contains a number restriction clash when adding assertions either by instantiating the mapping formula or by applying the propagation rules, then no canonical solution exists and the algorithm returns immediately with an empty solution. Otherwise, the algorithm terminates and generates a series of consistent ABoxes $\mathcal{A}_0, \mathcal{A}_1, ..., \mathcal{A}_n$. Starting from a consistent initial ABox $\mathcal{A}_0$, for the last ABox $\mathcal{A}_n$, we have the following property:

Lemma 1. *$\mathcal{A}_n$ is a solution.* □

Proof. (Sketch) $\mathcal{A}_n$ contains all assertions which can be constructed from the mapping formulas and tuples $\overline{s}$ such that $\mathcal{X} \models \psi(\overline{s})$. What we need to prove is that $\mathcal{A}_n$ is consistent. To do this, we will use the sound and complete tableaux algorithm for deciding $\mathcal{SHOIN}(\mathbf{D})$ knowledge bases. The algorithm is shown in the papers [4,5]. They show that if a knowledge base is satisfiable, then the algorithm does not generate any clashes.

The *propagation rules* used in our algorithm for computing a canonical solution is a subset of the *expansion rules* for deciding $\mathcal{SHOIN}(\mathbf{D})$ knowledge bases. However, in contrast to application of the *expansion rules*, our *propagation rules* only apply to individuals computed from the mapping formula and tuples in the XML document. Each individual has the form of $f_C(\overline{s})$.

Then it suffices to prove that if $\mathcal{A}_n$ does not contains the number restriction clash, then the decision procedure does not generate any clashes. □

Further, we have

Proposition 1. *$\mathcal{A}_n$ is a canonical solution. That is, given an XML document $\mathcal{X}$ and a CQ_0 query Q, $Q(\mathcal{A}_n) = \underline{certain}(Q, \mathcal{X})$.* □

Proof. Suppose Q is $q(\overline{x}) \leftarrow p_1(\overline{Y_1}),...,p_n(\overline{Y_n})$.

Lemma 1 has shown that $\mathcal{A}_n$ is a solution.

Let $\overline{s}$ be a tuple of data values. If $\overline{s} \in Q(\mathcal{A}_n)$, then there is a mapping h from $\overline{x}$ to $\overline{s}$ such that $(\mathcal{T}, \mathcal{R}, \mathcal{A}_n) \models p_i(h(\overline{Y_i}))$ for each i. We need to prove that for every solution $\mathcal{A}$, $\overline{s} \in Q(\mathcal{A})$, i.e., $(\mathcal{T}, \mathcal{R}, \mathcal{A}) \models p_i(h(\overline{Y_i}))$ for each i. Suppose a mapping formula Ψ is in the form $\psi(\overline{x}) \rightarrow \varphi(\overline{Y}, \overline{x})$:: *annotation*. Let $\mathcal{A}$ be a solution and let $\mathcal{I} = < \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} >$ be a model of $(\mathcal{T}, \mathcal{R}, \mathcal{A})$. Since $\mathcal{A}$ is a solution w.r.t. $\mathcal{X}$, then $< \mathcal{X}, \mathcal{A} >$ satisfies all Ψs in $\Sigma_{\mathcal{SO}}$, i.e., for a tuple $\overline{s'}$ in $\mathcal{X}$ such that $\mathcal{X} \models \psi(\overline{s'})$, there is a tuple $\overline{t}$ in Individual such that $\mathcal{A} \models \varphi(\overline{t}, \overline{s'})$ for each

formula $\varphi(\overline{Y}, \overline{x})$. By the construction of $\mathcal{A}_n$, we know that if there is a tuple $\overline{s'}$ in $\mathcal{X}$ such that $\mathcal{X} \models \psi(\overline{s}')$, then we add each assertion in the formula $\varphi(\overline{t}, \overline{s'})$ to $\mathcal{A}_n$ and $\mathcal{A}_n$ contains all and only the assertions which can be constructed from all mapping formulas $\varphi(\overline{t}, \overline{s'})$ and the axioms in TBox and RBox. Let $\mathcal{A}'_n$ be the set of assertions instantiated from all formula $\varphi(\overline{t}, \overline{s'})$ and $\mathcal{A}''_n$ be the set of assertions added by applying propagation rules. Since $\mathcal{A} \models \varphi(\overline{t}, \overline{s'})$ for each formula $\varphi(\overline{t}, \overline{s'})$, $\mathcal{A} \models \mathcal{A}'_n$. Hence, the model of $(\mathcal{T}, \mathcal{R}, \mathcal{A})$, $\mathcal{I} =< \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} >$, is a model of $(\mathcal{T}, \mathcal{R}, \mathcal{A}'_n)$. By the properties of the propagation rules, we know $< \mathcal{T}, \mathcal{R}, \mathcal{A}'_n > \models \mathcal{A}''_n$; therefore, $\mathcal{I}$ is a model of $(\mathcal{T}, \mathcal{R}, \mathcal{A}_n)$. Since $(\mathcal{T}, \mathcal{R}, \mathcal{A}_n) \models p_i(h(\overline{Y_i}))$ for each i, $\mathcal{I} \models p_i(h(\overline{Y_i}))$ for each i. By this we proved $(\mathcal{T}, \mathcal{R}, \mathcal{A}) \models p_i(h(\overline{Y_i}))$ for each i. Therefore, $\overline{s} \in \underline{certain}(Q, \mathcal{X})$. □

6 Conclusions

We have studied the problem of translating an XML document into an instance of an OWL-DL ontology. However, we are aware that the problem of checking the satisfiability of the ontology and the consistency of the mapping formulas as well as answering conjunctive queries still has a very high complexity – NEXPTIME-complete for OWL-DL ontologies and EXPTIME-complete for OWL Lite ontologies [6,9]. We attempt to investigate some efficient algorithms for answering conjunctive queries over OWL-DL ontologies, probably incomplete but acceptable, in the future.

Acknowledgments. We are grateful to anonymous reviewers for offering valuable comments, corrections, and suggestions for improvement.

References

1. Y. An, A. Borgida, and J. Mylopoulos. Constructing Complex Semantic Mappings between XML Data and Ontologies. In ISWC'05, 2005.
2. M. Arenas and L. Libkin. XML Data Exchange: Consistency and Query Answering. In PODS'05, Baltimore, USA.
3. R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data Exchange: Semantic and Query Answering. In ICDT'03.
4. I. Horrocks and U. Sattler. Ontology Reasoning in the SHIQ(**D**) Description Logic. In IJCAI'01.
5. I. Horrocks and U. Sattler. A Tableaux Decision Procedure for SHOIQ. In IJCAI'05.
6. I. Horrocks and S. Tessaris. A Conjunctive Query Language for Description Logic Aboxes. In AAAI'00.
7. E. Prud'hommeaux, A. Seaborne. SPARQL Query Language for RDF. http://www.w3.org/TR/2004/WD-rdf-sparql-query-20041012/
8. R. Rodriguez-Gianolli and J. Mylopoulos. A Semantic Approach to XML-bases Data Integration. In ER'01.
9. S. Tobies. Complexity Results and Practical Algorithm for Logics in Knowledge Representation. PhD Thesis, LuFG Theoretical Computer Science, RWTH-Aachen, Germany, 2001.

Self-tuning Personalized Information Retrieval in an Ontology-Based Framework

Pablo Castells[1], Miriam Fernández[1], David Vallet[1],
Phivos Mylonas[2], and Yannis Avrithis[2]

[1] Universidad Autónoma de Madrid, Escuela Politécnica Superior,
Campus de Cantoblanco, 28049 Madrid, Spain
{david.vallet, miriam.fernandez, pablo.castells}@uam.es
[2] National Technical University of Athens, School of Electrical and Computer Engineering,
GR-15773 Zographou, Athens, Greece
{fmylonas, iavr}@image.ntua.gr

Abstract. Reliability is a well-known concern in the field of personalization technologies. We propose the extension of an ontology-based retrieval system with semantic-based personalization techniques, upon which automatic mechanisms are devised that dynamically gauge the degree of personalization, so as to benefit from adaptivity but yet reduce the risk of obtrusiveness and loss of user control. On the basis of a common domain ontology KB, the personalization framework represents, captures and exploits user preferences to bias search results towards personal user interests. Upon this, the intensity of personalization is automatically increased or decreased according to an assessment of the imprecision contained in user requests and system responses before personalization is applied.

1 Introduction

Broadly speaking, information retrieval deals with modeling information needs, content semantics, and the relation between them [9]. Personalized retrieval widens the notion of information need to comprise implicit user needs, not directly conveyed by the user in terms of explicit information requests [7]. Again, this involves modeling and capturing such user interests, and relating them to content semantics in order to predict the relevance of content objects, considering not only a specific user request but the overall needs of the user.

When it comes to the representation of semantics (to describe content, user interests, or user requests), ontologies provide a highly expressive ground for describing units of meaning and a rich variety of interrelations among them. Ontologies achieve a reduction of ambiguity, and bring powerful inferencing schemes for reasoning and querying. Not surprisingly, there is a growing body of literature in the last few years that studies the use of ontologies to improve the effectiveness of information retrieval [5,8,10,11] and personalized search [4]. In this paper, we present a comprehensive personalized retrieval framework where the advantages of ontologies are exploited in different parts of the retrieval cycle: query-based relevance measures, semantic user preference representation, automatic preference update, and personalized result ranking. The framework is set up in such a way that the models benefit from each other

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 977 – 986, 2005.
© Springer-Verlag Berlin Heidelberg 2005

and from the common, ontology-based grounding. In particular, the formal semantics are exploited to improve the reliability of personalization.

Personalization can indeed enhance the subjective performance of retrieval, as perceived by users, and is therefore a desirable feature in many situations, but it can easily be perceived as erratic and obtrusive if not handled adequately. Two key aspects to avoid such pitfalls are a) to appropriately manage the inevitable risk of error derived from the uncertainty inherent to the automatic user preference acquisition by the system, and b) to correctly identify the situations where it is, or it is not appropriate to personalize, and to what extent. With this aim, our proposed framework incorporates a module to control the degree of personalization that is applied in the search result ranking, automatically adjusting it depending on the uncertainty contained in the search before personalization. The precision of ontology-driven semantics enables sharper observations within the system, upon which such uncertainty is assessed.

The rest of the paper is organized as follows. The semantic search framework is described in the next section. Section 3 explains the personalization model built on top of this framework. Section 4 is devoted to the techniques for the dynamic adjustment of the personalization effect. Section 5 describes our experimental setup for this system, after which some final remarks are given.

2 Ontology-Based Content Retrieval

Our ontology-based retrieval framework [11] assumes the availability of a corpus Δ of text or multimedia documents, annotated by domain concepts (instances or classes) from an ontology-based KB O. The KB is implemented using any ontology representation language for which appropriate processing tools (query and inference engines, programming APIs) are available. In our semantic search model, Δ rather than O is the final search space. Since the metadata attached to a document provide only, in general, a subset of the full document semantics, we advocate for a model of *imprecise semantic search*, where documents may satisfy a query to different degrees on a continuous scale (rather than a boolean relevance value), according to which they can be ranked.

Our retrieval model works in two phases (see Figure 1). In the first one, a formal ontology-based query (e.g. in RDQL) is issued by some form of query interface (e.g. NLP-based) which formalizes a user information need. The query is processed against the KB using any desired inferencing or query execution tools, outputting a set of ontology concept tuples that satisfy the query. From this point, the second retrieval phase is based on an adaptation of the classic vector-space Information Retrieval model, where the axes of the vector space are the concepts of O, instead of text keywords. Like in the classic model, in ours the query and each document are represented by vectors q and d, so that the degree of satisfaction of a query by a document can be computed by the cosine measure:

$$\mathrm{sim}(d,q) = \frac{d \bullet q}{|d| \cdot |q|}$$

The problem remains to build the d and q vectors, which is summarized next.

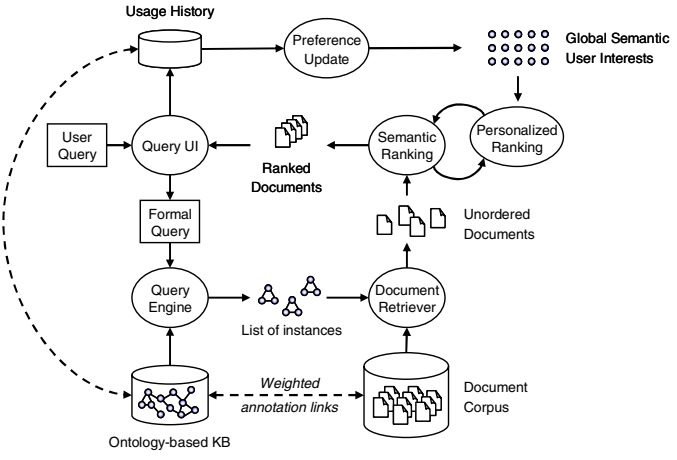


Fig. 1. The personalized ontology-based retrieval framework

Document vectors. Each content item in the search space Δ is represented by a vector d of concept weights, where for each domain concept $x \in O$ annotating d, d_x represents the importance of the concept x in the document (if x does not annotate d, then $d_x = 0$). The weight of annotations can be assigned by hand or automatically. If the document contains text, d_x can be computed automatically by a TF-IDF algorithm [9] as:

$$d_x = \frac{freq_{x,d}}{\max_y freq_{y,d}} \cdot \log \frac{|D|}{n_x}$$

where $freq_{x,d}$ is the number of occurrences of x in d, $\max_y freq_{y,d}$ is the frequency of the most repeated instance in d, and n_x is the number of documents annotated by x. This requires that an appropriate mapping of concepts to text keywords be available, whereby the number of occurrences of a concept in a document can be defined as the count of concept keywords in the text. What an appropriate mapping is in this context, and how it can be automated is a subject of active research [8].

For audiovisual documents, a variety of strategies can be used to weight the relevance of concepts in the content, based on automatic content analysis techniques, such as the size, movement, or relative position (e.g. foreground vs. background) of automatically recognized objects [2], measures of recognition certainty, text-based processing of speech transcripts, etc.

Query vector. The proposed construction of the query vector defines $q_x = 1$ if x appears in some tuple of the query result set, and 0 otherwise. The weights q_x can be further refined with a TF-IDF scheme, as suggested by [9]:

$$q_x = 0.5 + 0.5 \cdot \frac{freq_{x,q}}{\max_y freq_{y,q}} \cdot \log \frac{|D|}{n_x}$$

where we define $freq_{x,d}$ as the number of tuples of the result set where x occurs.

Our experiments confirm that this model outperforms keyword-based or image-based schemes, but not surprisingly degrades as the knowledge needed to answer queries is missing from the KB. Since it is not realistic in general to expect a complete coverage of the semantic space involved in large real-world document collections by means of domain KBs, our model is complemented with classic techniques to perform acceptably (i.e. not worse than standard retrieval techniques) when the knowledge is missing. Further details can be found in [11].

The proposed retrieval model takes advantage of the additional semantics (class hierarchies, precise and formalized relations) expressed by the ontology, that cannot be expressed using keywords. Moreover, it supports a notion of conceptual search based on fuzzy annotation of unstructured contents (text, media) by concepts, not supported by the traditional multifaceted search by document fields (e.g. title, author, date, etc.). Additionally, it provides elaborated tools for measuring the vagueness or uncertainty in the expression of an information need, as will be shown in Section 4.2, which will provide a way to assess the adequacy of personalizing the search and to what extent.

3 Ontology-Based Personalization

Personalization is a means to improve the performance retrieval (e.g. measured in terms of precision and relevance) as subjectively perceived by users [7]. The key aspects involved include the representation of user interests (beyond a specific one-shot query), the dynamic acquisition of such interests by the system, and the exploitation of user preferences. Our personalization framework is built as an extension of the ontology-based retrieval model described in the previous section. It shares the concept-based representation proposed for retrieval, and the expressiveness of ontologies to define user interests on the basis of the same concept space that is used to describe contents.

In our personalization framework, the semantic preferences of a user are represented as a vector $u \in [0,1]^{|O|}$ of concept weights, where for each domain concept $x \in O$, $u_x \in [0,1]$ represents the intensity of the user interest for x. With respect to other approaches, where user interests are described in terms of preferred documents, words, or categories, here an explicit conceptual representation brings all the advantages of ontology-based semantics, such as reduction of ambiguity, formal relations and class hierarchies. Our representation can also be interpreted as fuzzy sets defined on the sets of concepts, where the degree of membership of a concept to a preference corresponds to the degree of preference of the user for the concept. This interpretation is used in the definition of automatic preference extraction techniques based on the observation of user actions as is shown in the next section.

3.1 Automatic Preference Update

The approach followed for extracting user preferences for personalization is based on a formal methodology that is founded on fuzzy relational algebra and the existence of semantic relations amongst concepts. The extraction of preferences for semantic concepts is achieved by applying clustering algorithms on usage information data. The considered usage data consists of documents selected by the user for viewing them, or

explicitly marked as relevant in relevance feedback sessions. Our approach for extracting preferences from the history of user interaction consists of the clustering of documents based on the semantic annotation that matches concepts to documents, by which common topics implicit in clusters of concepts are detected.

The concept-vector representation of documents described in Section 2 can be reformulated to an equivalent interpretation of a document d as a normal fuzzy set on the set of concepts. Based on this set, and the knowledge contained in the form of available relations between the concepts, we aim to detect the degree to which a given document d is indeed related to a topic t. We will refer to this degree as $R(d,t)$. In other words, we attempt to calculate a relation $R : \Delta \times \mathrm{T} \rightarrow [0,1]$, where Δ is the set of available documents and T is the set of topics. Note that T is not known by the system beforehand, but emerges as a result of the algorithm. In designing an algorithm that is able to calculate this relation in a meaningful manner, three main issues need to be tackled. First of all, it is necessary for the algorithm to be able to determine which of the *topics* are indeed related to a given document, since a concept may be related to multiple, unrelated topics. In order for this task to be performed in a meaningful manner, the common meaning of the remaining concepts that annotate the given document needs to be considered as well. On the other hand, when a *document* is related to more than one, unrelated topics, we should not expect all the concepts that index it to be related to each one of the topics in question. Therefore, a clustering of concepts, based on their common meaning, needs to be applied. In this process, concepts that are misleading (e.g. concepts that resulted from incorrect annotation of the document) will probably not be found similar with other concepts that index the given document and therefore, the cardinality of the clusters may be used to tackle this issue. The main steps of the proposed algorithm are summarized in the following: (i) create a single relation that is suitable for use by thematic categorization, (ii) determine the count of distinct topics that a document is related to, by performing a partitioning of concepts, using their common meaning as clustering criterion, (iii) fuzzify the partitioning, in order to allow for overlapping of clusters and fuzzy membership degrees, (iv) take each cluster as a thematic topic and (v) aggregate the topics for distinct clusters in order to acquire an overall result for the document.

The topics that interest the user, and should be classified as positive interests are the ones that characterize the detected clusters. Degrees of preference can be determined based on the cardinality of the clusters, i.e., clusters of low cardinality should be ignored as misleading and the weights of topics in the context of the clusters, i.e., high weights indicate intense interest. The notion of high cardinality is modeled with the use of a *large fuzzy number* $L(\cdot)$, where $L(t)$ is the truth value of the preposition "the cardinality of cluster t is high". Therefore, each of the detected clusters t is mapped to positive interests by $u_x = \sum_{t \in T} \mu(x,t) \cdot L(t) \cdot K(t)$ for each $x \in \mathrm{O}$, where $\mu(x,t)$ denotes the degree of membership of the concept x to the cluster t, and $K(t) = \bigcap_{d \in t} R(d,t)$.

3.2 Personalization Effect

Once a semantic profile of user preferences is obtained, either automatically as described in the previous section, and/or refined manually, our notion of preference-based content retrieval is based on the definition of a matching algorithm that provides a personal relevance measure prm(d,u) of a document d for a user u. This measure is set according to the semantic preferences of the user, and the semantic annotations of the document, weighted as explained in Section 2. The procedure for matching d and u is based on a cosine function for vector similarity computation:

$$\mathrm{prm}(d,u) = \frac{d \bullet u}{|d| \cdot |u|}$$

In order to bias the result of a search (the ranking) to the preferences of the user, the measure above has to be combined with the query-based score without personalization sim(d,q) defined in Section 2, to produce a combined ranking. The combination of several sources of ranking has been the object of active research in the field of IR [3]. We have adopted the so-called combSUM model, by which the two rankings are merged by a linear combination of the relevance scores:

$$\mathrm{score}\ (d, q, u) = \lambda \cdot \mathrm{prm}\ (d, u) + (1 - \lambda)\ \mathrm{sim}\ (d, q)$$

where $\lambda \in [0,1]$. The choice of the λ coefficient in the linear combination above is critical and provides a way to gauge the degree of personalization, from $\lambda = 0$ producing no personalization at all, to $\lambda = 1$, where the query (local user interests) is ignored and results are ranked only on the basis of global user interests.

Given the inherent ambiguity of user actions upon which user preferences are automatically inferred, the automatic preference extraction techniques have an unavoidable risk of guessing wrong preferences, the negative effects of which increase with λ. Even when the extraction is most successful, there is considerable risk of contradicting explicit user requests if λ is too high, and λ should be therefore set with great care. It is commonly agreed that the user should have the means to turn personalization off (λ=0), or even tune λ as a free parameter (see e.g. Google Personalized[1]). Other than this, a fixed moderate value for λ can be typically set by experimental tuning, but we argue that the same λ is not necessarily appropriate for all situations. Further, we claim that it is possible to find hints in the context of a search according to which the level of personalization can be automatically self-adjusted, as is explained in the next section.

4 Gauging the Impact of Personalization

The degree to which the query dominates the retrieval process should vary in a manner that optimizes the retrieval result, i.e. in a manner that minimizes its uncertainty. As a general principle, if there is a high certainty that the results without personalization are relevant for an information need, personalization should be kept to a minimum. Put otherwise, the intensity of personalization should increase monotonically

[1] http://labs.google.com/personalized

with the degree of uncertainty in the search. Assessing (or even defining) such uncertainty with the information available in the system, before the results are presented to the user, is a fairly difficult problem in general. However, we propose an approximation to such assessment, by taking the *vagueness* in user requests and system responses as an approximation of the uncertainty in the search.

4.1 Assessing the Vagueness of the Search

In our proposal, the vagueness of a search (or equivalently, its *specificity* counterpart) is defined in terms of the specificity of the formal query, the query result set (concepts), and the final result set (documents), based on the retrieval model described in Section 2. Each of these three aspects is examined separately, and the corresponding results are combined into a single measure. More formally, given an ontology query q, containing a set of variables V_q, let $T_q \subset O^{|V_q|}$ be the result set of the execution of q (first retrieval phase as described in Section 2), and let $R_q = \{ d \in \Delta \mid \text{sim}(d, q) > 0 \}$ be the final result set in terms of documents (second retrieval phase). The specificity of a search is defined as a function $\text{spec}(q) = f(\text{spec}_1(q), \text{spec}_2(T_q), \text{spec}_3(R_q))$, where $f : [0,1]^3 \rightarrow [0,1]$ should be monotonically increasing with respect to its three variables. Our current empirical choice is the geometric mean $f(x, y, z) = (x \cdot y \cdot z)^{\frac{1}{3}}$.

The three partial specificity measures are defined as follows. First, we define $\text{spec}_1(q) = 1 - \frac{|V_q|}{3|C_q|}$, where $|V_q|$ is the number of variables in the query, and $|C_q|$ is the number of conditions. Thus, a query with many conditions and few variables is taken as more specific. Second, we take $\text{spec}_2(T_q) = 1 - \frac{\log(1+m)}{\log(1+|O|)}$, where $m = \left|\{x \in O \mid \exists t \in T_q, \exists v \in V_q, t_v = x\}\right|$, i.e. m is the number of distinct ontology elements occurring in the query result set. According to this measure, a query that is satisfied by many ontology instances (in relation to the size of the KB) is considered more unspecific. Finally, $\text{spec}_3(R_q) = 1 - \frac{\log(1+|R_q|)}{\log(1+|D|)}$, by which the fewer results a search returns, the more specific it is considered.

4.2 Adjusting Personalization by Impact

Once a notion of the vagueness of a query is established, a method for setting the degree of personalization in relation to that measure has to be defined. A very simple approach would be to set $\lambda = 1 - \text{spec}(q)$, which would implicitly take λ as a synonym for the "degree of personalization". But a better definition of "personalization impact" can be given in terms of the effective change of position of documents in the ranking. Dwork et al [3] propose the Spearman footrule distance to measure the difference between two search result lists as the average displacement of each document

across the rankings. We argue that a more significant measure of impact is whether or not a result will be seen at all by the user. Since in general the user will not browse the entire list of results, but stop at some top k in the ranking, there are a number of documents that the user would not see (the ones ranked after the k-th result) in the ranking without personalization, but would see as a result of a personalized reordering, and vice versa. If we count the rate of documents in the whole collection that cross the line for each possible value of k, and multiply it by the probability $P(k)$ that the user stops at each k, we get a loss function ranging in [0,1] that provides a measure of the effective impact (thus, the risk) of personalization in the retrieval process:[2]

$$\gamma(q,u) = \frac{1}{|D|}\sum_{k=1}^{|D|} P(k) \sum_{d \in D} \chi_k(d,q,u),$$

$$\text{where } \chi_k(d,q,u) = \begin{cases} 1 \text{ if } \text{sim}(d,q) \le k \text{ and } \text{score}(d,q,u) > k \\ 1 \text{ if } \text{sim}(d,q) > k \text{ and } \text{score}(d,q,u) \le k \\ 0 \text{ otherwise} \end{cases}$$

Now, rather than setting λ (i.e. the amount of personalization input) as a function of the vagueness of the search, we fix a desired output value for the effective impact of personalization in terms of that vagueness, and then set the value of λ that would yield this impact. Put formally, we equate $\gamma(q,u) = g(\text{spec}(q))$, and find λ from this equation, which is achieved as follows. To make $\gamma(q,u)$ linearly increasing with the uncertainty of the search, we define $g(\text{spec}(q)) = (1 - \text{spec}(q)) \cdot \gamma(q,u)|_{\lambda=1}$, where $\gamma(q,u)|_{\lambda=1}$ is the maximum value of $\gamma(q,u)$ for a given query, reached when $\lambda = 1$. $\gamma(q,u)$ is in fact a (monotonically increasing) function of λ, but it is not simple to invert this function analytically. However, it can be inverted empirically at runtime with high precision by computing $\gamma_i(q,u)$ for a discrete set of values $\lambda_i = i/n \in [0,1]$ with $1 \le i \le n$ (e.g. $n = 20$ which is not expensive[3]), and then defining $\lambda = \lambda_i$ for $\gamma_i(q,u) \le \gamma_n(q,u) \cdot (1 - \text{spec}(q)) < \gamma_{i+1}(q,u)$.[4] For the computation of $\gamma_i(q,u)$, we have taken an approximation to the distribution function for $P(k)$ by interpolation of data taken from a statistical study [6]. The final effect of this approach is that it is $\gamma(q,u)$, rather than λ, that is proportional to the vagueness of the search.

5 Early Experiments

We are testing our techniques on a corpus of documents from the CNN web site,[5] comprising 145,316 documents (445 MB). The domain ontology KB was taken from the KIM Platform [8], developed by Ontotext Lab,[6] with minor adjustments, plus the specific extensions of our framework (see [11] for a description of these). The domain KB includes 278 classes, 131 properties, 34,689 instances, and 462,848 sentences,

[2] We assume sequential browsing, i.e. the user does not see the i-th result before the $(i-1)$-th.
[3] Computing $\gamma_i(q,u)$ is $O(n \cdot m^2)$, where m is the number of elements in $\{d \in \Delta \mid \text{sim}(d,q) > 0 \text{ or } \text{prm}(d,u) > 0\}$.
[4] If $\gamma_i(q,u) = \gamma_{i+1}(q,u)$ for some i we would remove λ_i from the set without loss of precision.
[5] http://dmoz.org/News/Online_Archives/CNN.com
[6] http://www.ontotext.com/kim

taking a total of 70,5 MB in RDF text format (though in practice it is stored within a MySQL back-end using Jena 2.2). Based on the concept-keyword mapping available in the KIM KB, we automatically generated over $3 \cdot 10^6$ annotations (i.e. over 25 per document on average).

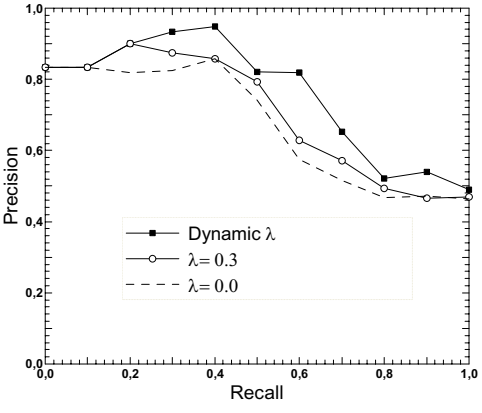


Fig. 2. The graphic illustrates the performance of our technique for the query "child organizations of public companies", using the standard precision vs. recall curve [9], according to different options to gauge personalization: i) personalization impact proportional to vagueness of the search, ii) personalization with fixed $\lambda = 0.3$, and iii) no personalization.

Figure 2 illustrates the performance of our techniques on the query "child organizations of public companies", compared to the results obtained without self-adjustment, and without personalization. In this case, the dynamic adjustment raises λ to 0.6 because the query is rather vague, according to the principles explained in Section 4.1, perceptively improving performance. The evaluation is done on the basis of a manual rating of document relevance on a scale from 0 to 5. Though our initial experiments are showing promising results, a more extensive testing is needed (and actually under way at the time of this writing), in order to complete and extend these first observations.

The experiments were run on a standard PC. Although systematic efficiency tests have not been conducted yet, the typical observed time to process a query takes below one minute. The main bottleneck is in the traversal of annotations (for the calculation of sim and prm), which are currently stored as an extension to the ontology. This cost grows linearly with the size of the result sets ($|T_q|$ and $|R_q|$). We expect to reduce this overhead by storing the annotations in a separate DB.

6 Conclusions

Reliability is a well-known concern in the field of personalization technologies. Since automatic preference modeling involves guessing implicit user's interests, it is impossible to approximate a total accuracy in meeting actual user needs. However it is possible to predict when the effect of potential failures can be serious, or close to

harmless. Our proposal aims at an automatic prediction of this effect, in order to raise or lower the level of personalization accordingly. The techniques are built upon a comprehensive framework that reaps the benefits from the expressive power and precision of ontologies in the different phases of the retrieval and personalization process.

The directions for the continuation of our work are manifold. To mention but a few, we are studying the integration of further ontology-based specificity measures (see e.g. [1,10]) in our uncertainty assessment techniques. Also, we plan to research a finer, qualitative, context-sensitive activation of user preferences, by which the level of personalization would not be uniform, but would be selectively distributed with a higher weight on the most context-relevant preferences.

Acknowledgements

This research was supported by the European Commission under contract FP6-001765 aceMedia. The expressed content is the view of the authors but not necessarily the view of the aceMedia project as a whole.

References

1. Anyanwu, K., et al: SemRank: Ranking Complex Relationship Search Results on the Semantic Web. 14th Intl. World Wide Web Conference (WWW 2005). Chiba, Japan (2005)
2. Bloehdorn, S., Petridis, K., Saathoff, C., et al: Semantic Annotation of Images and Videos for Multimedia. 2nd European Semantic Web Conference (ESWC 2005). Lecture Notes in Computer Science, Vol. 3532. Springer Verlag, Berlin Heidelberg New York (2005)
3. Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank Aggregation Methods for the Web. In Proc. of the 10th Intl. World Wide Web Conference (WWW10), Hong Kong (2001)
4. Gauch, S., Chaffee, J., and Pretschner, A.: Ontology-based personalized search and browsing. Web Intelligence and Agent Systems 1, Issue 3-4, IOS Press (2003) 219-234
5. Guha, R. V., McCool, R., Miller, E.: Semantic search. Proc. of the 12th Intl. World Wide Web Conference (WWW 2003), Budapest, Hungary (2003) 700-709
6. Jansen, B. J., Spink, A.: An Analysis of Web Documents Retrieved and Viewed. Proc. of the 4th International Conference on Internet Computing. Las Vegas, Nevada (2003) 65-69
7. Micarelli, A., Sciarrone, F.: Anatomy and Empirical Evaluation of an Adaptive Web-Based Information Filtering System. User Modelling and User-Adapted Interaction 14, Issue 2-3, Springer Science (2004) 159-200
8. Kiryakov, A., Popov, B., Terziev, I., Manov, D., Ognyanoff, D.: Semantic Annotation, Indexing, and Retrieval. Journal of Web Sematics 2, Issue 1, Elsevier (2004) 47-49
9. Salton, G. and McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)
10. Stojanovic, N., Studer, R., and Stojanovic, L.: An Approach for the Ranking of Query Results in the Semantic Web. In: Fensel, D., Sycara, K. P., Mylopoulos, J. (eds.): The Semantic Web – ISWC 2003, 2nd Intl. Semantic Web Conf. Lecture Notes in Computer Science, Vol. 2870. Springer Verlag, Berlin Heidelberg New York (2003) 500-516
11. Vallet, D., Fernández, M., and Castells, P.: An Ontology-Based Information Retrieval Model. 2nd European Semantic Web Conference (ESWC 2005). Lecture Notes in Computer Science, Vol. 3532. Springer Verlag, Berlin Heidelberg New York (2005) 455-470

Detecting Ontology Change from Application Data Flows

Paolo Ceravolo and Ernesto Damiani

Università di Milano, Dipartimento di Tecnologie dell'Informazione,
Via Bramante, 65, 26013 Crema, Italy
{ceravolo, damiani}@dti.unimi.it

Abstract. In this paper we describe a clustering process selecting a set of typical instances from a document flow. These representatives are viewed as semi-structured descriptions of domain categories expressed in a standard semantic web format, such as OWL [15]. The resulting bottom-up ontology may be used to check and/or update existing domain ontologies used by the e-business infrastructure.

Keywords: Knowledge Representation, Ontology Construction.

1 Introduction

Many successful electronic commerce applications use Web-based mediators to augment or replace human middlemen. Mediators organize data interchange based on high-level metadata, mapping domain resources to an interchange format; these metadata enable a variety of sharing, profiling, and querying services on domain data. Inter-organization metadata rely on a shared domain vocabulary whose terms are identified by an authority imposing a normative standard, often called *ontology*. The adoption of a standard ensures that the description of every resource will be written using terms provided by the authority via the shared vocabulary. However, this top-down process of metadata creation is often difficult to automate, and can be ineffective if the standard vocabulary does not cover all the semantic areas of interest for applications.

An alternative way that can be followed for producing a common vocabulary is *bottom-up* extraction of knowledge from business process data flow. A natural way of managing business process data flow is via web-services [7]. At this level, business processes can be described in terms of web-services providing functionalities and data. In business platforms based on web services, the data to be exchanged are described via XML Schema definitions. Individual SOAP messages contain data in XML format [13]; also, according to the ebXML standard for e-business, entire transactions are composed of business messages, whose payloads contain documents in XML [14] conforming to application-oriented XML Schemata.

One could therefore think of using these application-level XML schemata to extract high-level metadata; unfortunately, experience has shown that this

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 987–996, 2005.
© Springer-Verlag Berlin Heidelberg 2005

is rarely possible. XML Schema definitions used for business message payloads need to cover a wide repertoire of possible interchanges; therefore, they largely use structural mark-up, rather than describing data semantics. For this reason, we propose a technique for extracting metadata updates based on structural patterns detected in semi-structured XML data flow. Our clustering technique has been described in other works, such as [4] and [2]. In these papers, we focused on the clustering process selecting a set of typical instances from a document flow. In the present work, these representatives are viewed as semi-structured descriptions of a domain category, and organized as an ontology expressed in a standard semantic web format, such as OWL [15]. The resulting ontology is used to check and/or update existing domain ontology used by data mediators.

The paper is organized as follows: in Section 2 we address the problem of detecting instances from a data flow; in Section 2.1 we provide a brief discussion on motivations triggering changes on a domain ontology, and we specify the type of changes that are detected in our application; in Section 3 we describe the ontology construction process.

1.1 Related Work

Ontology building methodologies restricting their attention to semi-structured data integration are usually dealing with XML schemata. Schema are exploited as semantic definition of portion of information and integrated in a broaden representation. To be precise, schemata integration methodologies are not always semantic-aware. For example the MIX project as well as the Grammar Based Model, formalize integration rules on canonical tree-based models used to represent local DTD schemata and integrated schemata. Anyway in this work we are not interested in mere structural approaches.

The general strategy applied in semantic-aware approaches is to define an intermediate conceptual schema for mapping data to be integrated. The MOMIS project, propose an approach that merges, in a bottom-up way, structured and semistructured data sources. To achieve this, several rules and heuristics are applied in order to capture as much as possible the semantics of the elements of the DTDs. In the Clio project ([9]) data source schemata are transformed first into an internal representation, then, after the mappings between the source and the target schemas have been semi-automatically derived, the system materializes the target schema with the data of the source, using a set of internal rules, based on the mappings. DIXSE ([10]) follows a similar approach, transforming the DTD specifications of the source documents into an inner conceptual representation, with some heuristics to capture semantics. Most work, though, is done semi-automatically by the domain experts that augment the conceptual schema with semantics. The approach in [11] has an abstract global DTD, expressed as a tree, very similar to a global ontology. The connection between this DTD and the DTDs of the data sources is through path mappings: each path between two nodes in a source DTD is mapped to a path in the abstract DTD. Then, query rewriting is employed to query the sources.

2 Detecting New Types of Instances

In the remainder of the paper, we start from the assumption that the individual data items are XML fragments, complying with an application-level XML Schema. Also, we assume that this XML Schema is not intended to provide a partition of the domain instances into categories, and does not provide any hint on what "typical" data instances will look like. By mining the data flow, we shall detect recurrent types of domain instances, based on how individual XML elements are detailed in the number and type of sub-nodes as well as in the nodes content. Typical instances are extracted by clustering data of the transaction flow, and then we will use them to add new classes to our domain representation.

2.1 Ontology Changes

Evolving and updating a shared conceptualization such as an ontology is by no means an easy task, usually requires the knowledge of a qualified domain expert. Motivations triggering changes to ontology can be classified as follows:

- **Conceptualization tuning:** the new concept roughly represents the same concept as the old one, but does not have exactly the same instances. This can depend on concept scope, extension, or granularity.
- **Expression tuning:** depending on the relations among concepts, the same conceptualization can be expressed in different ways. For example, a distinction between two classes can be modeled using a qualifying attribute or by introducing a separate class.
- **Terminology tuning:** the same concept is described by means of synonyms or in terms of different encoding values (for instance distances can be expressed in miles or kilometers).

An extended discussion on ontology mismatches can be found in [12] and [3]. As we shall explain in detail in Section 3.2, in our approach new instance types will always comply with the application level XML Schema. For this reason, we shall focus on conceptualization tuning, i.e. on extension or granularity changes.

3 The Ontology Construction Process

In order to describe the individual steps of our ontology construction process we shall ground our discussion in an example. Fig. 1 shows an XML Schema of a generic purchase order services. This schema defines the message level of a Web-service based business transaction.

In the first step an initial set of *basic classes* are defined. This way a coarse-grained representation of the domain is provided. But this representation is not satisfiable and the classes are intended as *candidate classes*, to be detailed during the second step and eventually removed from the representation. This initial representation, although not very informative, can be created manually

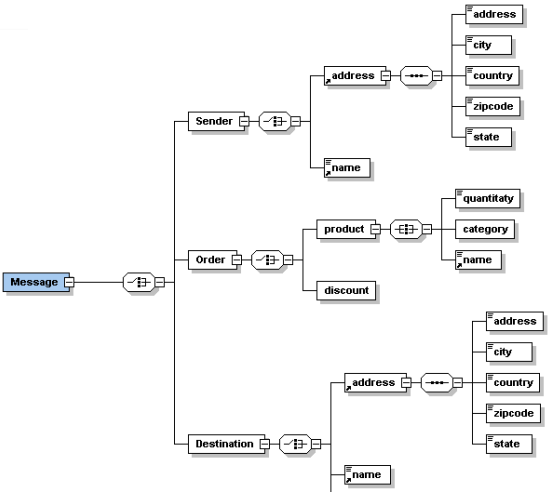


Fig. 1. An XML Schema of a generic purchase order

or easily extracted from the XML Schema used for messages payloads. We shall briefly describe this procedure in Section 3.1.

In the second step, new classes are induced by analyzing the data flow. Our clustering process dynamically partitions the flow in clusters of data items such as those in Fig. 3, each featuring a typical representative of the cluster. Then, we use such representatives to define new classes that can differ from the initial representation because of the cardinality of properties or because of the properties' values. For example `Message1` is a message without any order destination address, with an order composed of three products, and without an associated discount. These new classes can be expressed in terms of basic

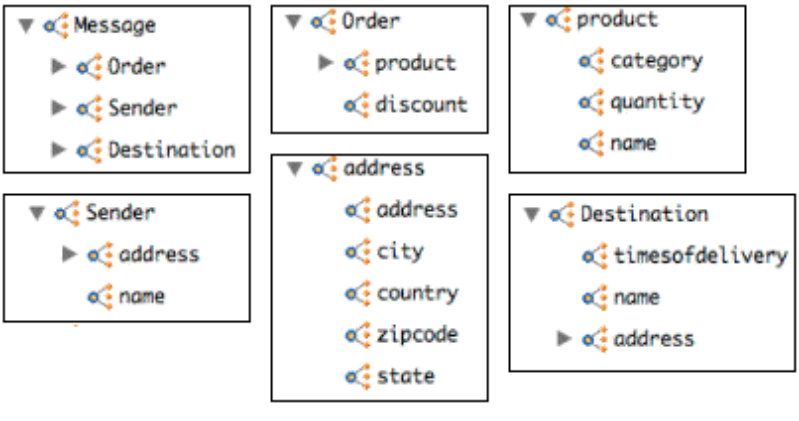


Fig. 2. Initial class induction

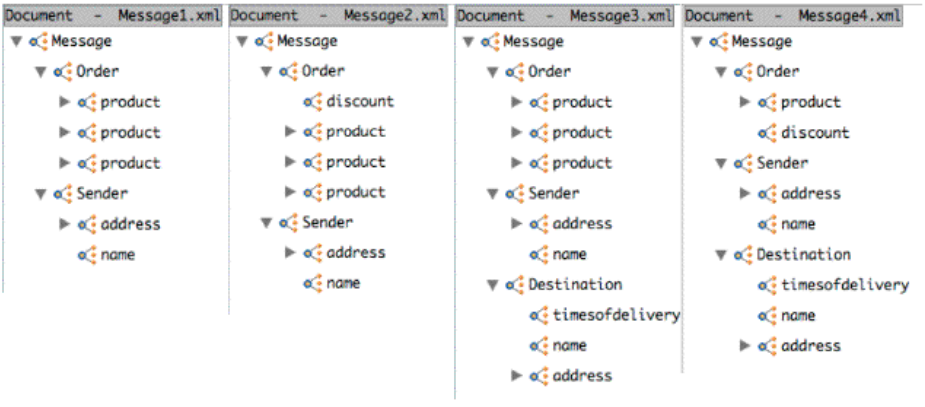


Fig. 3. Examples of clusters from the business transaction

classes, specifying a restriction on the class properties. In particular the restrictions we are interested to express are related to properties cardinality, and properties range type or values. For these reasons, the minimum language expressivity we need is a $\mathcal{ALQ}$ (a first order language restricted to formulas with two variables and allowing qualified restriction on roles), and OWL, that is a $\mathcal{SHIQ}(\mathcal{D})$ language cover these expressivity requirements [6]. In a naive approach, we could translate new induced classes directly in OWL, creating a new `Message` class constrained in the properties cardinality. But such a translation policy would not be effective, as it would produce one class per representative. For instance, `Message2` is a class very similar to the previous one, the only difference being the `Discount` property. For this reason, we adopt a *lazy* approach. Rather than transforming extracted information into change operations on our initial domain representation, we store them using an intermediate representation. At this intermediate representation level, *candidate classes* are maintained in a XML format: each candidate class is described via complex XML element composed of sub-nodes, as shown in Fig 2. This intermediate representation greatly simplifies the manipulation of the class hierarchy. More importantly, thanks to the intermediate layer the definition of change operations updating the ontology can be delayed to after the hierarchy gets stable.

3.1 Setting the Initial Ontology

Setting up an initial outline of a domain representation from scratch is a task at which human experts excel; also, it is known to be the fastest and least expensive phase of traditional domain modeling, and the most difficult to automate effectively [8]. As an alternative, if domain metadata such as database or XML schemata are available, quick-and-dirty automatic construction of a preliminary domain model in basic classes can be done by reverse engineering[1]. Specifically, when a reasonably well-behaved application-level XML Schema is

available, it is just a matter of enumerating its elements belonging to complex types[1]. Then, XML elements can be easily translated into OWL by transforming each XML element into a class having as name the element name and as properties the sub-elements. If a sub-element contains its own sub-elements, the corresponding property is declared a `owl:ObjectProperty` and takes as range a class having the sub-element name. If a sub-element is a leaf (i.e., it does not contain sub-elements), the property is declared as a `owl:DatatypeProperty`. Classes produced by means of this process are inserted into the intermediate representation and are to be intended as candidate classes to be confirmed only after the optimization of the hierarchy maintained in the intermediate representation. Fig. 2 shows a sample initial ontology created from a partition of the XML Schema. Its classes provide an initial representation of the domain, to be detailed and updated by means of knowledge extracted from the data flow.

3.2 Detecting Detailed Class Descriptions

We are now ready to use extracted knowledge to improve the quality of the initial model. Fig. 3 shows some examples of clusters extracted from the e-business data flow. These clusters are specific instantiations of the schema of the service messages. Note that clusters can be composed only of valid schema elements: a complex XML element may occur or not in a clusters, but if it does it necessarily complies with the schema. Obviously, XML elements belonging to `SimpleType`s can contain different values.

In order to extract knowledge from our clusters, we start by partitioning them in basic classes. Then we compare basic classes with existing candidate classes, and decide whether the cluster is providing new candidate classes; if so, they are added to the set of existing candidates. For example, by partitioning `Message1` we obtain two new basic classes. We call the first class `Message-I`: it differs from the original `Message` because of the `destination` element. We call the second `Order-I`: it is an `Order` having three `product` elements and no `discount`. The `Sender` class does not differ from the initial Sender class extracted from the schema, and therefore provides no new candidate. Fig. 4 shows in a graphical way the new basic classes extracted from the flow in Fig. 3.

Using new basic classes `Message1` can be described as a `Message-I` class having as sub-nodes `Order-I` and `Sender`. Note that evaluating complex elements we only take into account direct sub-elements. This prevents overproduction of classes definitions.

This way, new classes are expressed in terms of restrictions on the initial domain representation.

Assertion 1 shows how we can express this information according to a Description Logic formalism.

[1] For the sake of conciseness, we shall not explain in detail here the naming conventions to be observed in XML elements and `ComplexType` s declarations for a XML Schema to be considered well-behaved w.r.t. knowledge representation. Unfortunately, these conventions are not always followed in real world applications.

$$\textit{Message-I} \sqsubseteq \textit{Message}$$

$$\textit{Message-I} \sqsubseteq \forall \textit{order}.\textit{Order-I} \quad (1)$$

The new class is described by means of a set of complex class definitions. Currently we use two types of definitions:

- *specialization*, specifying that the extracted class is a subtype of a class already present in the domain
- *restriction*, specifying that the extracted class is obtained from a class already present in the domain by restricting its properties cardinality or range.

Note that we interpret specialization as necessary (although not sufficient) relation between the newly defined class and the class used in the definition. In fact we declared `Message-I` as a sub class of `Message` (instead of defining it as an equivalent class) because we have a more detailed vision on the instances description. Also, we used a universal operator for expressing the restriction on the property `order`, because the cluster is intended to be a representative of a set of transactions, and is description must to be valid for each instance.

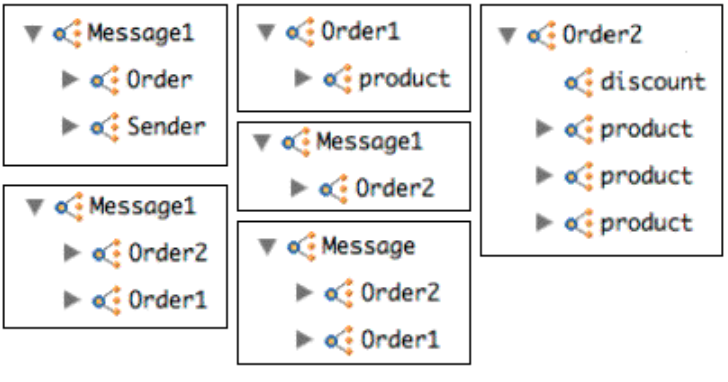


Fig. 4. The new basic classes extracted from the flow

The above example shows how the new class increases the knowledge embedded in the initial domain representation, and tunes it to the specific application setting.

3.3 Setting up the Ontology Hierarchy

Periodically, our system uses available candidate classes to set up or update a hierarchy. This can be done simply connecting each class to all the other classes it subsumes. These operations are made on the intermediate representation that simplify our manipulation task. Classes are connected be means of `is-a` relations,

and the basic task we have to execute is to decide when a class have to be connected with another. A class A is said to be `is-a` a class B when it is more general.

For our purposes, we cannot use a standard DL reasoning engine. DL reasoners are based on algorithms for subsumption resolution, where the generality of an assertion is done in comparison to a set of axioms. In our application an assertion is said to be more general of another on the basis of its definition. We define `is-a` as follows:

Definition 1. *A class A is said be* `is-a` *a class B if A contains less elements than B and all the elements contained in A are also contained in B.*

It is easy to see that our definitions generates a number of redundant closure arcs: if a class A `is-a` B, it also `is-a` all other classes in `is-a` relation with B. But this information is maintained only in the intermediate representation, when the hierarchy is translated in OWL these redundant information are not exported. Fig. 5 shows a hierarchy obtained the definition 1, for the sake of clarity redundant relations are removed.

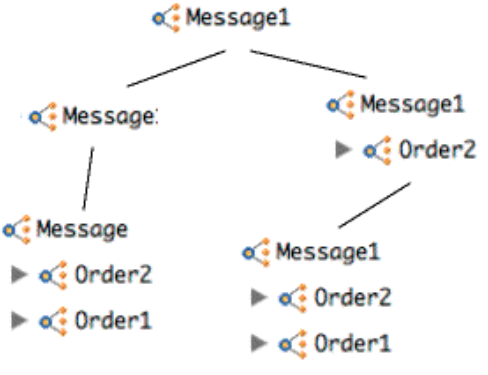


Fig. 5. The first hierarchy applied to the definitions of clusters

The last phase of the process is the optimization of the hierarchy. If multiple classes share the same set of sub-classes, the hierarchy can be simplified. As stated in Section 3.2 (and shown in assertion 1) a cluster is defined as a complex class, i.e. a class defined by means of other class. or property restrictions. We deal with single complex assertions as class properties and use definition 1 for managing subsumption. The optimization process starts from more specific class definitions, and is managed in 3 step:

- find, if exists, an equal class definition;
- compare with other definitions and create an `is-a` relation if the definition is more general;
- insert the class definition in a list for avoiding further controls.

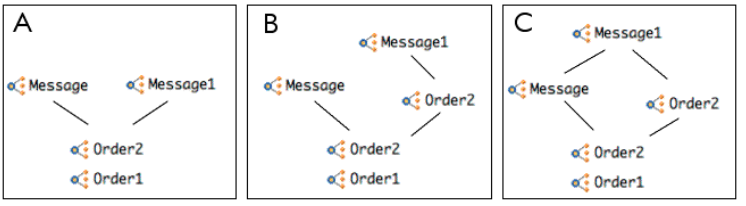


Fig. 6. An example of optimized hierarchy

Note that once the process is performed on a class definition more general definitions have not to take into account this class during the second step, i.e. the `is-a` relation setting. This way, the computational complexity of the process is strictly related to the number of class definition to be evaluated.

Fig. 6 shows the application of optimization steps to the hierarchy in Fig.5. The overall effect is to reduce the number of assertion lines, because classes sharing the same assertions are factored out, i.e. defined as super-classes of a sub-class composed by the shared assertions.

Note that simplification of the hierarchy does not involve changes in the conceptualization, but only in the model expression. The expressive power of a simpler hierarchy is exactly the same of a more structured one. Differences concern only metadata production and querying: a more structured ontology allows for producing a complex metadata assertion in less steps then a simpler ontology, and can require for less complex query for accessing data.

For these reasons, in our approach the user can choose whether to execute or not a hierarchy optimization, according to the application requirements.

4 Conclusions

In this paper, we have presented some techniques enabling extraction of knowledge from XML data. Our approach is aimed at extracting hierarchies of typical XML messages (i.e., typical data items) from a flow of business transactions. First, we cluster data items building an intermediate knowledge representation; then, our intermediate representation is lazily compared to an initial normative schema, obtaining a more detailed specification of the domain, expressed as OWL complex class definition. We intend to develop this approach toward a compete bottom-up approach for building ontology schema on the basis of e-business data interchange, capable of checking and/or updating existing domain ontologies used by the e-business infrastructure.

Acknowledgments

This work was partly funded by the Italian Ministry of Research Fund for Basic Research (`FIRB`) under projects `RBAU01CLNB_001` “Knowledge Management for the Web Infrastructure” (`KIWI`). and `RBNE01JRK8_003` “Metodologie Agili per la Produzione del Software” (`MAPS`).

References

1. Andersson M.: Extracting an Entity Relationship Schema from a Relational Database through Reverse Engineering. LNCS, vol 881, proceedings of the 13th International Conference on the Entity-Relationship Approach, (1994).
2. Ceravolo P., Nocerino M. C., Viviani M.: Knowledge extraction from semi-structured data based on fuzzy techniques. Eighth International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES 2004), Wellington, New Zealand, (2004) 328–334.
3. Chalupsk H.: OntoMorph: A translation system for symbolic logic. In Cohn, A. G., Giunchiglia F., Selman B., editors, KR2000: Principles of Knowledge Representation and Reasoning, San Francisco, CA. Morgan Kaufmann. (2000) 471–482.
4. Damiani E., Nocerino M. C., Viviani M.: Knowledge Extraction from an XML Data Flow: Building a Taxonomy based on Clustering Technique. EUROFUSE Workshop on Data and Knowledge Engineering (EUROFUSE 2004), Warszawa, Poland, (2004) 22-25.
5. Heflin J., and Hendler J.: Dynamic ontologies on the web. In Proceedings of the Seventeenth National Conference on Artificial Intelligence. AAAI/MIT Press, Menlo Park, (2000) 443–449.
6. Horrocks I., Sattler U., and Tobies S.: Practical reason- ing for very expressive description logics. J. of the Interest Group in Pure and Applied Logic, 8(3), (2000) 239–264.
7. Koushik S., Joodi P.: E-Business Architecture Design Issues, IT Professional, vol 2, num 3 IEEE Educational Activities Department, Piscataway, NJ, USA, (2000) 38–43.
8. Maedche A., Staab S.: Ontology Learning for the Semantic Web , IEEE Intelligent Systems, (2001).
9. Popa L., Velegrakis Y., and Miller R.J.: Translating Web Data, In the proceedings of VLDB02, (2002) 598–609.
10. Rodriguez-Gianolli P., and Mylopoulos J.: A Semantic Approach to XML-based Data Integration. In ER, volume 2224 of Lecture Notes in Computer Science, (2001) 117–132.
11. Reynaud C., Sirot J.P., and Vodislav D.: Semantic Integration of XML Heterogeneous Data Sources. In IDEAS, IEEE Computer Society, (2001) 199–208.
12. Visser P. R. S., Jones D.M., Bench-Capon T. J. M., Shave, M. J. R.: An analysis of ontological mismatches: Heterogeneity versus interoperability. In AAAI 1997 Spring Symposium on Ontological Engineering, Stanford, USA, (1997).
13. SOAP Version 1.2 W3C Recommendation 24 June 2003, `http://www.w3.org/TR/soap/`
14. ebXML SPECS `http://www.ebxml.org/specs/`
15. OWL Web Ontology Language Overview, W3C Recommendation 10 February 2004 `http://www.w3.org/TR/2004/REC-owl-features-20040210/`

SeBGIS 2005 PC Co-chairs' Message

Nowadays new applications ask for enriching the semantics associated to geographical information in order to support a wide variety of tasks including data integration, interoperability, knowledge reuse, knowledge acquisition, knowledge management, spatial reasoning and many others. Examples of such semantic issues are temporal and spatio-temporal data management, 3D manipulation, spatial granularity and multiple resolutions, multiple representations (providing different perspectives of the same information), vague and ambiguous geographic concepts, the relationship between geographic and physical concepts, and identity of geographic objects through time.

At the same time the recent years brought many developments that radically changed how we understand information processing. Data warehouses and OLAP systems have evolved as a fundamental approach for developing advanced decision support systems. This lead to improved data mining techniques allowing to extract semantics from raw data. Further, the success of Internet has generated a paradigm shift in distributed information processing leading to the area of Semantic Web, in which semantics is the fundamental component for achieving communication both for humans and applications. At the same time, mobile and wireless computing have entered everyone.s life through dedicated devices leading to location-based services. Finally, Grid computing, a paradigm enabling applications to integrate computational and information resources managed by diverse organizations in widespread locations, pushes the frontier of global interoperability. The fact that all these recent developments are entering the geographic domain increases the importance of the elicitation of the semantics of geographical information.

The aim of this workshop is to bring together researchers from academia and industry as well as practitioners for discussing views on how to integrate semantics into current geographic information systems, and how this will benefit the end users. The workshop will be organized in a way to highly stimulate interaction amongst the participants. To this aim, an important part of each regular time slot will be reserved for a discussion.

Three experts from the same or a closely related discipline as the authors reviewed each of the thirty-four submissions. The program committee and the additional experts who willingly took on the burden of careful review made a strong contribution to the quality of the workshop. Eighteen papers have been chosen. These papers are of high quality and promise a successful and fruitful workshop.

The papers will be presented into five Sessions. Session 1 (day 1) - with four papers - presents how semantics may be measured, evaluated and enriched. Session 2 (day 1) - with three papers - focuses on the schemata integration. Session 3 (day 1) - with four papers - contributes to various topics related to geovisualization and spatial semantics. Session 1 (day 2) - with three papers - focuses on the algorithms and data structures for semantic based Geographic Information

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 997–998, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Systems. Session 2 (day 2) - with four papers - presents a set of specialized systems and tools. The workshop will be concluded with Session 3 (day 2), which will be devoted to an open discussion (round table) on the outcomes of this workshop and the future of Semantic-based Geographic Information Systems.

We wish you all a pleasant and fruitful workshop!

August 2005

Esteban Zimányi, Université Libre de Bruxelles
Emmanuel Stefanakis, Harokopio University of Athens
(SeBGIS'05 Program Committee Co-Chairs)

How to Enrich the Semantics of Geospatial Databases by Properly Expressing 3D Objects in a Conceptual Model

Suzie Larrivée[1,2,3], Yvan Bédard[1,2,3], and Jacynthe Pouliot[1,2]

[1] Dept of Geomatics Sciences
[2] Centre for Research in Geomatics
[3] Canada NSERC Industrial Chair in Geospatial Databases for Decision Support,
Laval University, Quebec City, Canada
Telephone: 1-418-656-2131; Fax: 1-418-656-7411
{suzie.larrivee, yvan.bedard, jacynthe.pouliot}@scg.ulaval.ca

Abstract. Geospatial conceptual data models represent semantic information about the real world that will be implemented in a spatial database. When linked to a repository, they offer a rich basis for formal ontologies. Several spatial extensions [5, 15, 17] have been proposed to data models and repositories in order to enrich the semantics of spatial objects, typically by specifying the geometry of objects in the schema and sometimes by adding geometric details in the repository. Considering the success of such 2D spatial extensions as well as the increased demand for 3D objects management, we defined a 3D spatial extension based on the concept of PVL already used in Perceptory and elsewhere. This paper presents 3D concepts and 3D PVL to help defining the geometry of 3D objects in conceptual data models and repositories. Their originality stems from the fact that no similar solution exists yet for real-life projects. The enrichment of the meaning of 3D objects geometries is discussed as well as its impact on costs, delays and acquisition specifications.

1 Introduction

At the beginning of 70's, the Entity-Relationship Model has emerged to represent semantic information about the real world that implementation model cannot do [7, 8, 9]. Based on this conceptual formalism, many researchers [5, 15, 17] have worked to extend E/R with pictograms to represent and enrich the semantics of 2D spatial objects. Such conceptual data models, with their repository (or dictionary), have been used in several projects and proved useful to describe the semantics of spatial objects stored in spatial databases, including their geometric characteristics. In the most recent solutions, such extensions are used with UML (Unified Modeling Language).

Increasingly, 3D objects management is becoming a common requirement in spatial database systems [1, 6]. Nevertheless, 3D characteristics of objects are still poorly depicted in database conceptual schemas and repositories. Following the successful use of a 2D PVL (Plug-in for Visual Language) extension with UML and of the integration of a rich repository for 2D spatial database models, we enriched the developed PVL with 3D elements to better define the spatial characteristics of 3D objects and consequently have more meaningful database contents. This paper presents this new set of PVL pictograms which can be used to better define the

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 999 – 1008, 2005.
© Springer-Verlag Berlin Heidelberg 2005

geometry of 3D objects in a conceptual data model and repository. We first define 3D concepts that help to reduce semantic confusion. Afterward, we present the developed solution to improve the semantics of 3D objects geometries and discuss the importance of properly describing the geometric meaning of 3D objects. We focus on the conceptual level and voluntarily do not go deep in the underlying concepts, hoping to help clarifying concepts that still remain widely confused in scientific literature. The sole diversity of meanings that still exists for "3" and "D" hampers the proper use of 3D concepts by practitioners. In addition, formal meta-modeling and concepts related to levels of modeling, multiple representations, spatial and temporal relationships, generalisation, constraints and human cognition have been discussed in previous papers or technical reports and they underly the present paper.

2 Fundamental 3D Concepts

There exist different definitions of 3D objects. Often, there is confusion between the dimensions of the object shape and the dimensions of the space in which these objects are located. For example, according to ESRI, a three-dimensional shape is: "*a point, line, or polygon that stores x-, y-, and z-coordinates as part of its geometry. A point has one set of z-coordinates; lines and polygons have z-coordinates for each vertex*" [10]. Such definition appears semantically incorrect because it does not refer to the number of dimensions of the object shape (point 0D, line 1D or polygon 2D) but to the number of dimensions needed to locate these objects in a 3D universe. The definition given by Euclid's Elements[1] web site presents a different view: "*A solid is that which has length, breadth, and depth*". Such view defining a 3D shape as a solid avoids confusion between the number of dimensions of the universe and those of the object, as it is the case in 2D topology with 0-cell, 1-cell and 2-cell objects. Mathworld Web site[2] proposes a good definition: "*the dimension of an object is a topological measure of the size of its covering properties. Roughly speaking, it is the number of coordinates needed to specify a point on the object. For example, a rectangle is two-dimensional, while a cube is three-dimensional. The dimension of an object is sometimes also called its "dimensionality"*." In other words, the number of dimensions of an object is the number of coordinates necessary to uniquely locate a point in this object: 0 in a point, 1 in a line, 2 in a polygon and 3 in a solid. This is the definition that we have adopted as it is mathematically more robust.

Such definition also implies that objects can serve as a universe to locate other objects. It is the case for example with roads which are 1D objects usually located in a 2D or 3D universe but which can also be used as a 1D linear referencing system (LRS) to locate other objects like accidents, road signs, etc. The next paragraphs clarify the concepts of universe dimensions and objects dimensions.

2.1 Dimensions of a Universe

The number of dimensions of a universe corresponds to the number of spatial axis (or coordinates) needed to uniquely locate objects in this universe. For example, a 2D

[1] http://aleph0.clarku.edu/~djoyce/java/elements/elements.html
[2] http://mathworld.wolfram.com/Dimension.html

universe has 2 axes and a 3D universe has 3 axes. Objects located in a universe cannot have more dimensions than the universe used to locate this object except if this universe is itself an object located in another universe having more dimensions. For example, a 2D parking can be located at the offset of a 1D road between two points on this road (the road being a 1D universe located in a 2D universe).

2.2 Dimensions of Objects

In this paper, object means an element or feature of the reality represented by a shape in a spatial universe. The number of dimensions of this object follows Mathworld' mathematical definition, i.e. it is the number of axes needed to locate a point within this object when it is used as a universe. It is based on the space occupied by the object itself (e.g. length, width, thickness) and not the space occupied by its minimum bounding rectangle (MBR) or minimum bounding box (MBB) which are usually defined parallel to the coordinate axes in a universe with more dimensions. Accordingly, a line is a 1D object whether it is a straight, curvilinear (included in a 2D MBR) or a non-planar line (included in a 3D MBB).

3 3D Database Modeling with 3D PVL Expressions

Nowadays spatial database applications ask for enriching the semantics associated to geographical objects to support a wide variety of tasks such as data integration, interoperability, knowledge reuse and spatial reasoning. It is the role of conceptual data model, as Chen says [6], “to incorporate some of the important semantic information about the real world”. It also is their role to contribute to building a formal semantics. Semantics has varying meanings in sciences like Linguistics, Philosophy, Anthropology and Artificial Intelligence. In this paper, we use the definition given to semantics by Logic Science, that is “the study of relationships between signs and symbols and what they represent”[18]. In cartography and 3D modeling, signs and symbols are combined with geometry to convey a meaning to what we see, to help recognize objects. Visual variables (ex. color, line weight, line style, patterns) bear meaning and explicitly take part to the semantics of objects. The components of spatial reference (position, shape, size and orientation) can be neutral but can also bear meaning and then contribute to semantics. For example, on a 2D paper map, red points, red rectangles and red detailed polygons can be interpreted as small, medium and large buildings or as residential, commercial and public buildings according to the legend of this map. The legend adds meaning to the shape of objects while geometry allows one to infere spatial relationships which are meaningful for the understanding of a phenomena. In digital maps, some meaning of the geometric feature comes from its name, identifier and attributes. However, these are not sufficient to understand the complete meaning beared by a geometry. One may ask "what types of buildings are represented by points, rectangles and detailed polygons? Do points represent residential buildings, small buildings or both? Do polygons represent public buildings, buildings larger than 200m² or both? Are roads all of the same width or are they symbolized? etc." It is possible that such meaning isn't explicitely stored into attributes or cartographic layers or object classes but can only

be deduced from the geometry and symbology. Consequently, geometric definitions stored in repositories to describe data acquisition specifications as well as the derivation rules (ex. generalization) are important to understand the meaning of these geometries and of the objects they represent. Table 1 shows a semantic table adapted to spatial databases where values are geospatial objects (in columns) and geometric categories (in row) which combinations correspond to the semantic of features. The first column is the genus (semantic group with common attributes) and the other columns are the differentia (attributes that serve to distinguish genus from each others). In conceptual data schema, the geometric category of objects is represented with PVL pictograms (explained in the next section) while the differentia allowing to distinguish each building type are described in the repository as specifications.

Table 1. Semantic table combining geometry and cartographic semiology with Genus and Differentia to distinguish different categories of buildings

	Genus	Differentia	
	Building	Public	Small
Point	+	-	+
Rectangle	+	-	-
Polygon	+	+	-

Now, let's suppose that roads are part of a 3D spatial database. What is a road in the context of this spatial database? At very large scale, which part of the road is used to digitize the polygon representing its boundaries: the main pavement ? the shoulders? cadastral boundaries? Is there a minimum length to create a new road? What is the granularity of the polygonal information, does-it include all small variations of its width at intersections? Is it the real shape or a simplified shape? How does-it start and ends at intersections? Does it cover all the roads of the city or only those connected to numbered roads? etc. Having the geometric description associated to this object class contributes to enrich the semantics of the objects by stating that the polygons representing the paved roads correspond to the limits of their pavement, that only roads longer than 250m and larger than 15m are digitized, that polygons don't include geometric changes smaller than 10% of the width of the road, and so on. Thus, in this specific example, shorter roads are not considered to be a road, minor width modifications are not represented, etc. It is thus important to define geometry in a conceptual model and a repository to better describe the meaning or semantic of objects and better understand the impact on spatial relationships.

To help database designers to describe the geometry of spatial objects, we developed the concept of Plug-in for Visual Language (PVL) that helps describing the geometry of object classes in a conceptual schema and a repository. Such PVLs are technology-independent and visual modeling-language independant, they help the analyst to better describe what users want without worrying about implementation issues. The next section presents the 3D PVL and the repository forms used to describe the geometry in detail. The PVL grammar rules are presented after.

3.1 PVL Pictograms and Their Repository Forms

PVL is a language composed of a small number of signs (called pictograms) and of a few grammar rules. PVL is meant to act as an extension to any modeling language and as such, is a specialized language of its own. PVL expressions, made of pictograms and rules, can represent visually in a conceptual schema the geometry and temporality of objects classes, attributes and processes of a spatiotemporal database. The three fundamental spatial pictograms of the PVL for a 2D universe are ⊡, ⊿ and ⬟ which designate respectively a 0D object , a 1D object and a 2D object. The two fundamental temporal pictograms are ◷ and ◔ respectively for instants (0D) and intervals (1D). Several articles have been written to describe these pictograms and the grammar used to combine them to describe complex geometries, complex temporalities and spatio-temporal objects [1, 3, 4, 5], they are not repeated here.

A specialized repository is always joined to the schema to detail PVL expressions when needed. In addition to including most of the semantic information found in ontologies, this repository contains information about digitizing objects or acquisition specifications, processes to derive objects' geometry and temporality, sources for geometry or temporality, etc. (Fig. 1). Metadata about spatial reference systems and quality information are stored in additional forms.

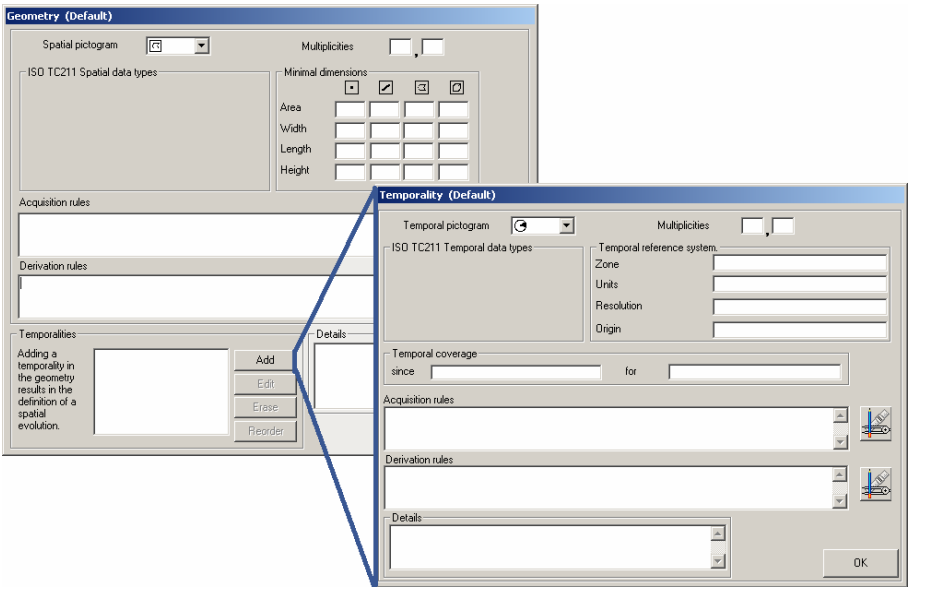


Fig. 1. Perceptory's repository forms used to enrich the semantics of objects and to detail the mapping specs in addition to storing their spatial and temporal pictograms

Recently, we added new pictograms to the PVL for 3D conceptual database modeling [1, 13], which differ from 2D spatial pictograms by showing a box to include the geometry instead of a square. This box represents the geometry of the

universe (3D) that includes the shape of the object class. Explicitely depicting the number of dimensions of the universe in which is located an object has become necessary with the possibility of the most recent technologies to store several objects from different data sources in a same data warehouse or to produce different views in different universes (ex. Oracle Spatial). It has also become necessary with the increased importance of multi-representation databases.

To obtain the shape of the objects located in a 3D universe, we transpose each shape of the 2D spatial pictograms in the 3D box in a way that preserves their ground trace (2D aerial view) and that gives them a thickness (or elevation) or not. We obtain the six pictograms shown in Table 2.

Table 2. How a 2D geometry can be transposed in a 3D universe

Objects in a 2D universe	become in a 3D universe:	
	flat objects or objects draped on a DTM	objects with height or thickness
		
		
		

The PVL can be used with any CASE tools because its pictograms are included in a font. They are also used with Perceptory, a CASE tool developed especially for geospatial databases. In all cases, the number of dimensions of the universe and of each object class is depicted visually as explicitly as possible.

3.2 Grammar Rules to Combine Pictograms into More Complex 3D Geometries

The grammar rules used to generate the appropriate 3D PVL expressions are described in the next paragraph using the EBNF (Extended Backus-Naur Form) standard ISO/IEC 14977 [10].

Table 3. Used EBNF notation

Symb.	Meaning	Symb.	Meaning
=	Defining-symbol	( )	Start and End-group-symbol
,	Concatenate-symbol	[ ]	Start and end-option-group
|	Definition-separator-symbol	{ }	Start and End-repeat-symbol
’	Quote symbol	;	Terminator-symbol
(* *)	Start and End-comment-symbol	-	Exclusion-symbol

Hereafter, we present the rules concerning only the 3D PVL although it is possible to combine 3D pictograms with 2D pictograms in the case of multiple-representations spatial databases or with the temporal pictograms for spatiotemporal databases. It is also possible to use them for attributes and methods the same way as the other pictograms.

3DPicto = (3DSimplePicto | 3DComplexPicto), [Multiplicity];
3DSimplePicto = '' | '' | '' | '' | '' | '';
3DComplexPicto = '' | '' | '' | '' | '' |

semantically the same features but they depict different 3D definitions. In the conceptual object classes of Fig. 3, x,y and z coordinates are meant to be measured in 3D for each object class. Trees are intended to be vertical lines, walls to be vertical plans, and buildings aggregates of plans (i.e. not full solids). In Fig. 3, the desired geometries are meant to be measured in a 2D universe. Then, 3D-like geometries are derived through two processes: one to give the bottom-z, one to give the top-z to those objects having an attribute 'height'. This will give, for example, buildings with flat roofs.

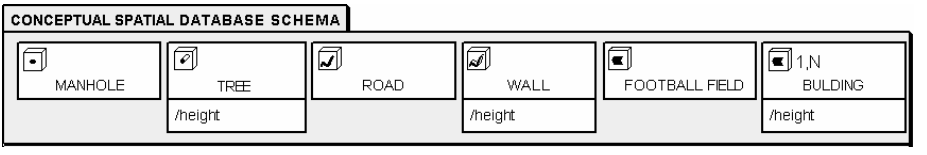


Fig. 3. Subset of a conceptual 3D data schema (not involving associations and methods)

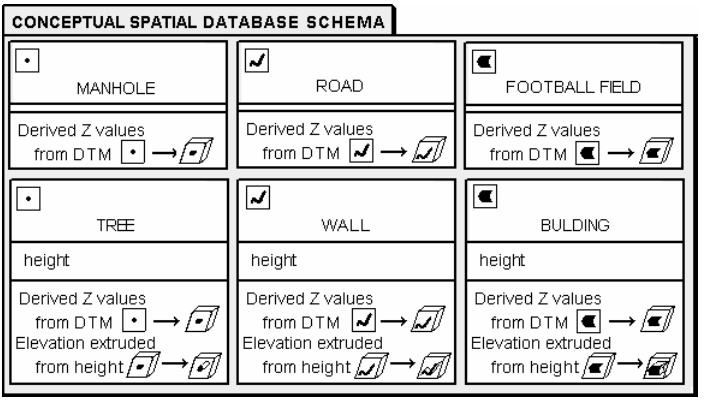


Fig. 4. Conceptual 3D data schema where objects are initially meant to be 2D but where Z-base can be derived from a DTM and Z-top from an attribute height

Such PVL expressions clearly highlight the concerns about the manipulation and analysis of the third dimension, thus helping to choose the best 3D software. For example, some GIS are only 2D and are limited to generate Grid and TIN. Other GIS offer 2½D environments, they can only have one z coordinate for each x,y pair and consequently do not support solids and don't allow perfectly vertical lines and plans (one needs to create microscopic offsets in order to have 2 z values for practically the same x,y pair). “Current GIS 3-D representation does not truly exist. Many existing GIS models are actually modeling a 2.5-D environment.” [18]. Using such strategy allows this category of GIS to add dimensionality to a 2D universe and give thickness to flat objects. Another alternative is to store x, y and z coordinates for each vertex as is supported by CAD systems or some DBMS, thus offering 3D.

These examples highlight the different meanings, geometrically speaking, that 3D objects may have. They also highlights the need to describe 3D objects in a way that meaningfully depicts what they are intended to be, independant from implementation.

Deciding the type of 3D objects a user truly needs requires a good understanding of the possibilities since the final choice about the intended 3D geometries has a major impact on the cost, delay and complexity of data acquisition, data processing and software selection. To properly express 3D needs, one can to use the proposed 3D pictograms in a way that meaningfully expresses these needs. This conceptual step is more delicate than in 2D because of the largest impact. We need to say more than "I want 3D objects", we need to have explicit meanings for 3D geometries. More meaningful 3D geometric information helps the user to interpret the true sense of objects to be included in a map or 3D model. When needed, details can be stored in the repository in natural or formal language, leading to richer 3D ontologies and better understanding of potential 3D relationships between objects.

5 Conclusion

The semantic of objects has been defined for several years with conceptual models and repositories. Including geometric definitions contributes to semantics. While spatial extensions for modeling languages have existed for over 15 years, nothing specific to 3D objects and universes existed insofar. The proposed 3D PVL paired with a repository allows one to define the subtelties of 3D geometry. It allows the analyst to create a conceptual database model that depicts a clear understanding of the several issues concerning 2D vs 2½D vs 3D, acquisition vs derivation of Z, 2D-thick objects vs true 3D-shape objects, etc. It allows one to see more clearly what he can expect from the database (ex. volumetric and 3D topological analysis) and the spatial relationships he can infere from it. In other words, it adds meaning to data models in a way that is cognitively compatible with most users and systems analysts.

Aknowledgements

The authors wish to acknowledge the financial support of Canada NSERC, Laval University, R&D Defense Canada. We are thankful to the users who send us feedbacks. We also thank the partners of the NSERC Indutrial Chair in geospatial database for decision support: Hydro-Quebec, RDDC, Intelec, KHEOPS, Syntell, DVP, Holonics, Transport Quebec, Natural Resources Canada.

References

1. Arens, C., Stoter, J., van Oosterom, P.: Modelling 3D objects in a Geo-DBMS using a 3D primitive. Computers & Geosciences, 31: (2005) 165-177.
2. Bédard, Y., Larrivée, S., Proulx, M-J., Nadeau, M.: Modeling Geospatial Databases with Plug-Ins for Visual Languages: A Pragmatic Approach and the Impacts of 16 Years of Research and Experimentations on Perceptory. In: S. Wang et al. (Eds.): Conceptual Modeling for Advanced Application Domains. Lecture Notes in Computer Science, Vol. 3289, Springer-Verlag, Berlin Heidelberg New York (2004) 17–30

3. Bédard, Y., Pouliot, J., Larrivée, S., Frenette, P., Normand, P., Brisebois, A.: Création d'un modèle 3D urbain de la recherche de données à l'exploitation du modèle 3D. Research Report realised for Defence Research and Development Canada (2002)
4. Bédard, Y.: Visual Modelling of Spatial Database towards Spatial PVL and UML. Geomatica, 53(2) (1999) 169-185
5. Bédard, Y., Paquette F.: Extending entity/relationship formalism for spatial information systems. AUTO-CARTO 9, April 2-7, Baltimore (1989) 818-827
6. Billen, R. Nouvelle perception de la spatialité des objets et de leurs relations. Développement d'une modélisation tridimensionnelle de l'information spatiale. Ph.D. Theses, Université de Liège, Faculté de Sciences, Département de géographie, 2002.
7. Chen, P. P.: The Entity-Relationship Model - Toward a Unified View of Data. ACM Transactions on Database Systems (TODS), vol 1, n° 1 (1976) 9-36
8. Deheneffe, C., Hennebert, H., Paulus, W.: Relational model for data base. Proc. IFIP Congress, North-Holland Pub. Co., Amsterdam (1974) 1022-1025
9. Hainaut, J.L., Lecharlier, B.: An extensible semantic model of data base and its data language. Proc. IFIP Congress, North-Holland Pub. Co., Amsterdam (1974) 1026-1030.
10. ISO/IEC 14977: Information technology -- Syntactic metalanguage -- Extended BNF (1996)
11. Karman, M., Amdahl, G.: Dictionary of GIS terminology. ESRI Press (2001)
12. Kraus, K.: Photogrammetry: Advanced Methods and Applications, Dümmler-Verlag (1997)
13. Larrivée, S., Bédard, Y., Pouliot, J.: Modélisation conceptuelle des bases de données géospatiales pour des applications 3D. Revue internationale de géomatique, numéro spécial: Information géographique tridimensionnelle: théories, systèmes et applications (2006)
14. Molenaar, M.: A formal data structure for 3D vector maps. Proceedings of EGIS'90, Amsterdam, The Netherlands (1990) 780-781
15. Parent, C., Spaccapietra, S., Zimanyi, E., Donini, P.: Modeling Spatial Data in the MADS Conceptual Model. Int. Symp. on Spatial Data Handling, Vancouver (1998) 138-150
16. Schmid, H. A., Swenson, J. R.: On the semantics of the relational model. Proc. ACMSIGMOD, Conference, San Jose, Calif. (1975) 211-233
17. Shekhar, S., Vatsavai1, R.R., Chawla, S., Burk, T. E.: Spatial Pictogram Enhanced Conceptual Data Models and Their Translation to Logical Data Models. ISD'99, LNCS, Vol. 1737, Springer Verlag, Berlin Heidelberg New York (1999) 77-104
18. The American Heritage Dictionary of the English Language. Houghton Mifflin (1981)
19. Thurston, J.: Geo-Visualisation : Current Issues / Future Potentials. GISCafe.com (2001)

Evaluating Semantic Similarity Using GML in Geographic Information Systems

Fernando Ferri[1], Anna Formica[2], Patrizia Grifoni[1], and Maurizio Rafanelli[2]

[1] IRPPS-CNR, via Nizza 128, 00198 Roma, Italy
{fernando.ferri, patrizia.grifoni}@irpps.cnr.it
[2] IASI-CNR, viale Manzoni 30, 00185 Roma, Italy
{formica, rafanelli}@iasi.cnr.it

Abstract. This paper proposes a method for evaluating the semantic similarity of GML elements (concepts). Due to the relevance of the *Is-in* relationship in the geographic context, it focuses on GML elements organized according to *Part-of* (*meronymic*) hierarchies. It also proposes the method's application to *Part-of* hierarchies, due to the semantics of the meronymic relationship within the geographic context, referred to as "place-area". This semantics essentially concerns parts which are similar to and inseparable from the whole. A further contribution refers to the modeling of the *Part-of* hierarchy in GML. In particular, from the perspective of applying the *information content* approach to *Part-of* hierarchies, this paper proposes a method to represent and distinguish the concepts involved in the meronymic relationship within the element hierarchy.

1 Introduction

In Geographic Information Systems (GISs) semantic similarity plays an important role, as it supports the identification of objects that are conceptually close, but not identical. It is acquiring particular importance in the retrieval of geospatial data in settings such as heterogeneous databases, digital libraries and the World Wide Web, where users have different backgrounds and no precise definitions of the subject matter [1]. A semantic similarity model facilitates comparison of entities and allows information retrieval and integration to handle semantically similar concepts . The goal of a similarity model is to obtain flexible and better matches between user-expected and system-retrieved information.

GML (Geography Markup Language) is emerging as the dominant standard for exchanging geographic data across the Internet [2]. For this reason, we propose a method for evaluating the semantic similarity of GML elements (concepts). Given the relevance of the *Is-in* relationship in the geographic context, we focus on GML elements organized according to *Part-of* (*meronymic*) hierarchies [3]. Semantic similarity of hierarchically related concepts has been widely investigated in the literature [4] [5]. From the various proposals, we followed the probabilistic approach of [6], which is based on the notion of *information content* and overcomes the drawbacks of the traditional *edge-counting* approach. Note that the information content approach, originally introduced by Resnik in [7], was conceived to compare concepts within *Is-a* hierarchies. Here, we propose its application to *Part-of* hierarchies, due to the semantics of the meronymic relationship within the geographic context, referred to as "place-area" [8]. This semantics essentially concerns parts which are similar to and inseparable from the whole.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1009 – 1019, 2005.
© Springer-Verlag Berlin Heidelberg 2005

A further contribution of this paper concerns the modeling of the *Part-of* hierarchy in GML. To our knowledge, in GML the meronymic relationship is represented by using the native element hierarchy of the language [9] [10], therefore by dealing with parts as attributes. In this paper, from the perspective of applying the information content approach to *Part-of* hierarchies, we propose a method to represent and distinguish the concepts involved in the meronymic relationship within the element hierarchy.

The paper is organized as follows: in Section 2, we examine some important works proposed in the literature. In Section 3 we explain how to code a *Part-of* Hierarchy with GML, and in Section 4 we address the problem of evaluating the similarity of GML elements. Finally, Section 5 concludes.

2 Related Works

Similarity is a widely used, important concept. Semantic similarity in particular plays a significant role in GISs, as it supports the identification of objects that are conceptually close, but not identical. The problem of how to evaluate semantic similarity has been addressed by various authors at many times and from different points of view.

A natural way to compute semantic similarity in a taxonomy is to evaluate the distance between the nodes corresponding to the items being compared (the shorter the path from one node to another, the more similar they are). For example, in [11] and [12] the authors suggest that the assessment of similarity in semantic networks can in fact be thought of as involving just taxonomic (*Is-a*) links, to the exclusion of other link types. Then, distance-based measures of concept similarity assume that the domain is represented in a network, but such measures are not applicable if a collection of documents is not represented as a network. However, an acknowledged problem with this approach, also referred to the edge-accounting approach, is that in real taxonomies links do not generally represent uniform distances.

For this reason in [4], [6], [7] semantic similarity in an *Is-a* taxonomy is measured on the basis of the information content (that is, also the approach followed in this paper). In particular, [4], [7] propose algorithms that take advantage of taxonomic similarity in resolving syntactic and semantic ambiguities. For instance, in [4] the author affirms that "semantic similarity represents a special case of semantic relatedness: for example, cars and gasoline would seem to be more closely related than, say, cars and bicycles, but the latter pair are certainly more similar." He also affirms that links, such as *Part-of*, can also be viewed as attributes that contribute to similarity (see also [13], [14]). In [6] the author investigates an information-theoretic definition of similarity that is applicable as a probabilistic model. The similarity measure is not directly stated as in earlier definitions but derived from a set of assumptions, in the sense that if the assumptions are accepted, the similarity measure necessarily follows. He demonstrates how his definition can be used to measure the similarity in a number of different domains. He also shows that this proposal can be used to derive a measure of semantic similarity between topics in an *Is-a* taxonomy, and briefly discusses the different points of view mentioned above tied to particular applications or to particular domain models.

In [5] the authors present an approach to computing semantic similarity that relaxes the requirement of a single ontology and accounts for differences in the levels of

explicitness and formalization of the different ontology specifications. A similarity function determines similar entity classes. Experimental results with different ontologies indicate that the model gives good results when the ontologies contain complete and detailed representations of entity classes. In [1] the same authors define the *Matching-Distance Similarity Measure* for determining semantic similarity among spatial entity classes, taking into account their distinguishing features (parts, functions, and attributes) and semantic interrelations (*Is-a* and *Part-whole* relations). Always from a *spatial* point of view, in [15] the authors discuss the need for semantic integration and present a prototype of an information source integration tool, which focuses on schema integration of spatial databases. This tool is able to recognize the similarities and the differences between entities to be integrated. A domain-dependent ontology is created from the Federal Geographic Data Committee and domain-independent ontologies (Cyc and Wordnet). The authors use a ratio model (ontology nodes distance) to assess the similarities and differences between terms.

In [16] the authors distinguish between *tree-based similarity* and *graph-based similarity*. The former starts from the notion that the information content of a class or *topic t* (or a concept c) is defined as – *log p(t)* (the above mentioned probabilistic model); i.e. as the probability of a concept increases, the informativeness decreases, therefore the more abstract a concept the lower its information content, as also affirmed in [17]. The latter generalizes the *tree-based similarity measure* to exploit both the hierarchical and non-hierarchical components of an ontology. The proposed graph-based semantic similarity measure was applied to the Open Directory Project ontology. The described methodology to evaluate ranking algorithms based on semantic similarity can be applied to arbitrary combinations of ranking functions stemming from text analysis.

In [18] the author combines *link* and *content analysis* to estimate semantic similarity. The paper reports on the first attempt to approximate semantic associations by mining content and link information from billions of pairs on Web pages.

Other proposals are made in [19], where *Dice* and *Cosine* coefficients are proposed, (even if they are applicable only when the objects are represented as numerical feature vectors) and [20], where the authors investigate the idea of finding semantic similarity between search engine queries based on their *temporal* correlation and develop a method of efficiently finding the highest correlated queries for a given input query using far less space and time than the naïve approach.

3 Coding a Part-of Hierarchy with GML

The real world in the geographic domain can be represented as a set of features. AbstractFeatureType codifies a geographic feature in GML. Each feature represents a specific concept in the geographical domain. Here, we use the term concept to point to a geographic feature of GML, such as elements or types. Geographic concepts with geometry are those with properties that may be given a geometrical value. Geometry type is an important property, included in the reference coordinate system and describing the extent, position or relative location of the represented concept. A reference system provides a measurement scale for assigning values “to a location, time or other descriptive quantity or quality”. GML codifies the fundamental geometric types in geometry.xsd. The most important correspond to the following

geometric types: PointType, MultiPointType, LinearRingType, LineStringType, MultiLineStringType, MultiCurveType, PolygonType, MultiPolygonType, MultiSurfaceType, MultiGeometryType.

The geometric types defined in GML provide the framework for modelling all geographical concepts. This framework enables modeling, for example, of the concepts composing a network of communication ways, such as roads, rivers, canals and other communication infrastructures. Figure 1 shows an example of a type hierarchy that introduces concepts concerning communication infrastructures starting from the GML geometric types. The concepts introduced in the example are defined as an extension of the MultiLineStringType.

As mentioned in the Introduction, due to the relevance of the *Is-in* relationship in the geographic context, this paper focuses on GML elements organized according to *Part-of* (*meronymic*) hierarchies. For instance, in our example a *Part-of* relationship exists among Communication Ways (ComWay) and roads, rivers and canals. In the literature, *Part-of* hierarchies are usually modelled in XML using "sequences of elements" [9] [10]. In fact, an element may contain subelements, which may in turn contain other subelements. A similar approach could be followed in GML.

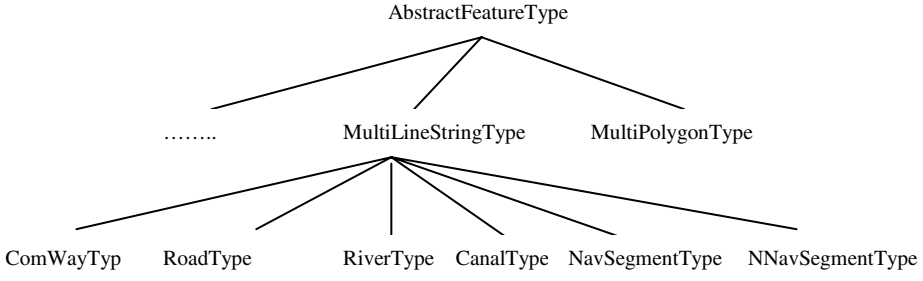


Fig. 1. Type Hierarchy

According to it, in Figure 2 ComWay represents a composite element, consisting of three component elements: Road, River and Canal. However, with this approach it is not possible to distinguish between elements of the *Part-of* hierarchy and any other defined elements (out of the *Part-of* hierarchy) such as Kind and Country.

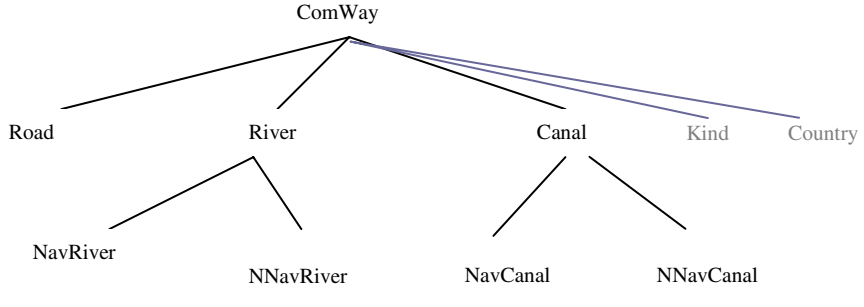


Fig. 2. Element Hierachy

To highlight meronymic relationships within the GML element hierarchy, a *Part-of* hierarchy could be modelled by introducing some special geographic types such as the PartOfWayType, PartOfRivType, PartOfCanType of Table 1. Each special type is introduced to model a *Part-of* relationship between a geographic concept and their component concepts. This approach enables, for example, representation that the geographic concepts Road, River and Canal are *Part-of* the geographic concept

Table 1. The Communication ways (ComWay) example

```
<element name="ComWay"
      type="ComWayType"/>
<element name="Road" type="RoadType"/>
<element name="River" type="RiverType"/>
<element name="Canal" type="CanalType"/>
<element name="NavRiver"
      type="NavSegmentType"/>
<element name="NNavRiver"
       type="NNavSegmentType"/>
<element name="NavCanal"
      type="NavSegmentType"/>
<element name="NNavCanal"
       type="NNavSegmentType"/>

<complexType name="ComWayType">
 <sequence>
   <element name = "kind" type="string"/>
   <element name = "country" type="string"/>
   <element name = "PartOfWay"
       type="PartOfWayType"/>
 </sequence>
 <attribute name="label" type="string" />
 <attribute name="label" type="string" />
 <attribute name="length" type="integer" />
</complexType>

<complexType name="PartOfWayType">
 <sequence>
   <element name = "Road" type="RoadType"/>
   <element name = "River" type="RiverType"/>
   <element name = "Canal" type="CanalType"/>
 </sequence>
</complexType>

<complexType name="RoadType">
   <attribute name="label" type="string" />
   <attribute name="length" type="integer" />
   <attribute name="maxspeed" type="integer" />
</complexType>

<complexType name="RiverType">
 <sequence>
   <element name = "PartOfRiv"
      type="PartOfRivType"/>
 </sequence>
 <attribute name="label" type="string" />
 <attribute name="length" type="integer" />
 <attribute name="flow" type="integer" />
 <attribute name="deepness" type="integer" />
</complexType>
```

```
<complexType name="CanalType">
 <sequence>
   <element name = "PartOfCan"
       type="PartOfCanType"/>
 </sequence>
 <attribute name="label" type="string" />
 <attribute name="length" type="integer" />
 <attribute name="capacity" type="integer"/>
 <attribute name="deepness" type="integer"/>
</complexType>

<complexType name="PartOfRivType">
 <sequence>
   <element name = "NavRiver" type="
       NavSegmentType "/>
   <element name = "NNavRiver" type="
       NNavSegmentType"/>
 </sequence>
</complexType>

<complexType name="PartOfCanType">
 <sequence>
   <element name = "NavCanal"
       type="NavSegmentType"/>
   <element name = "NNavCanal" type="
       NNavSegmentType"/>
 </sequence>
</complexType>

<complexType name="NavSegmentType”>
 <attribute name="id" type="string" />
 <attribute name="length" type="integer"/>
 <attribute name="maxspeedVehicle"
      type="integer"/>
 <attribute name="season" type="boolean"/>
 <attribute name="people" type="boolean"/>
 <attribute name="goods" type="boolean"/>
</complexType>

<complexType name="NNavSegmentType”>
 <attribute name="id" type="string" />
 <attribute name="length" type="integer"/>
 <attribute name="usage" type="string"/>
</complexType>
```

ComWay, by a special geographic concept *Part-of* ComWay (PartOfWay), and NavigableRiver (NavRiver), and NonNavigableRiver (NnavRiver) as *Part-Of* River, etc.

Using this approach it is possible to represent the *Part-of* hierarchy of Figure 3.

Table 1 shows a GML example of ComWay, including the element hierarchy represented in Figure 3.

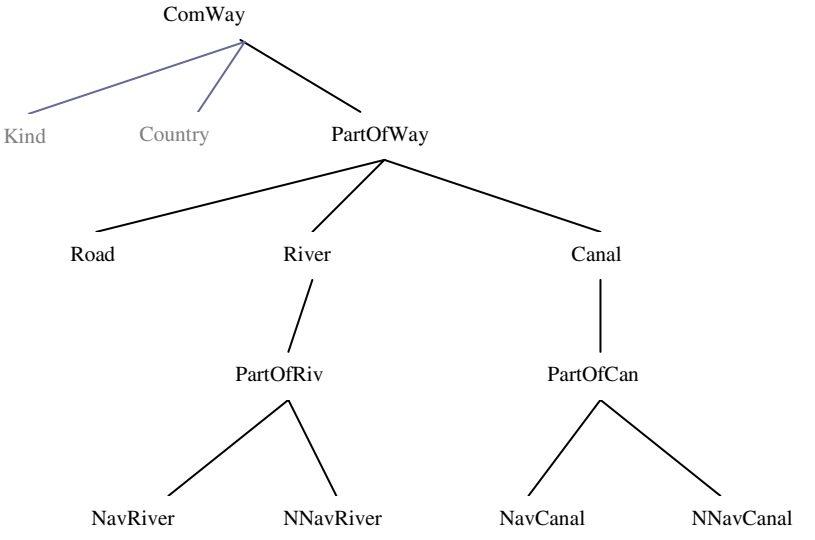


Fig. 3. Element Hierarchy evidencing parts

By summarizing, to define the parts in GML, the proposed approach adds evidence types (elements) specifying "*Part-of* " properties. It should be clear that a taxonomy mixing parts with the subclass hierarchy of Figure 1 or elements not specifying parts, such as Country and Kind in Figure 2, leads to semantic ambiguities.

4 Evaluating Similarity

The problem of evaluating the similarity of GML elements is addressed below. The proposed method was inspired by a previous proposal [21], which presented a method for measuring XML-Schema element similarity. In the mentioned paper the similarity of element and attribute names is axiomatically defined by making use of domain *ontologies*, while here we have adopted the *information content* approach defined by Lin [6], originally introduced by Resnik [7]. The starting assumption of the method is the association of weights with the taxonomy concepts, representing the probability of encountering instances of those concepts along the hierarchy. As we focus on *Part-of* taxonomies, we therefore concentrate on the concepts of the element hierarchy that are the children of "*Part-of* nodes". The association of probabilities with the *Part-of* taxonomy allows us to introduce the notion of a *weighted element hierarchy*. In our example, probabilities have been estimated in line with WordNet 2.0 [22]. For instance, the concepts *Road* and *River* are defined below, with their relative frequencies (the numbers in parentheses). Probabilities are therefore computed by

dividing frequencies by the total number of concepts in the taxonomy (50,000 in the case of WordNet).

(95) Road - an open way (generally public) for travel or transportation;
(55) River - a large natural stream of water (larger than a creek).

The weighted element hierarchy of our running example is shown in Figure 4. Note that the *Kind* and *Country* concepts are not weighted, as they are not involved in the *Part-of* hierarchy.

Given a weighted taxonomy, the information content of a concept c is defined as $-\log p(c)$; i.e. as the probability of a concept increases, the informativeness decreases, therefore the more abstract a concept, the lower its information content [17]. Therefore, according to [4] and [7], the similarity of concepts is given by the maximum information content shared by the concepts: the more information two concepts share, the more similar they are. Of course, the least upper bound of the concepts, if it exists, provides the maximum information content shared by the concepts. In our approach, given two concepts, the notion of *information content similarity* (*ics*) between concepts is introduced. It provides the maximum information content shared by the concepts in the weighted element hierarchy, according to [6]. In addition, if we assume the existence of a set of sets of *synonyms* in the given domain (for instance, as defined according to WordNet 2.0), and c_1, c_2 are two concepts belonging to one of these sets, than $ics(c_1, c_2) = 1$. Of course, for any concept c, $ics(c,c) = 1$, and in all other cases, the *ics* is null.

For example, consider the *River* and *Canal* concepts. According to the probabilities shown in Figure 4, the following holds:

$$ics\,(River,Canal) = 2\,logp(ComWay)/(logp(River) + logp(Canal)) = 0.72.$$

Note that, according to Lin's approach, either the maximum information content shared by the concepts or the information contents of the concepts to be compared are considered. This is not true if we adopt the pure Resnik's approach which addresses only the maximum information content shared by the concepts. For instance, according to Resnik, the similarity between *River* and *Canal* coincides with that of *Road* and *Canal*, both pairs having the same maximum information content (i.e. that provided by *ComWay*).

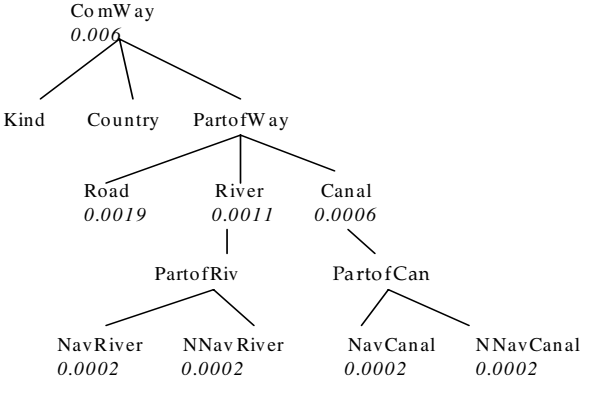


Fig. 4. Weighted Element Hierarchy of Communication Ways (ComWay)

For complexTypes, the comparison is performed according to an extension of a previous proposal for evaluating concept similarity in an ontology management system called *SymOntos* [23]. This is inspired by the *maximum weighted matching* problem in bipartite graphs. It was conceived to compare both the sequences of elements (not involved in the *Part-of* hierarchy) and the sets of attributes of the complexTypes. Due to lack of space, we focus on the sets of attributes only, as they are the relevant component in the description of the complexTypes of our running example. The method is briefly summarized as follows. Consider the sets of attributes of two complexTypes, and all the sets of pairs of attributes, each pair formed by attributes not belonging to the same complexType, such that no two pairs of attributes share an element. For instance, if we consider two sets of attributes representing a set of boys and a set of girls, a possible set of pairs is defined by a possible set of marriages (when polygamy is not allowed) [24]. Then, (one of) the set of pairs is selected such that the sum of the *ics* is maximum. For instance, consider the types associated with *River* and *Canal*, *RiverType* and *CanalType*. In this case the set of pairs of attributes that maximizes the sum of the related *ics* is the following:

{(*label*,*label*), (*length*,*length*), (*flow*,*capacity*), (*deepness*,*profundity*)}

since, by assuming that *deepness* and *profundity* are synonyms, we have:

ics(*label*,*label*) = *ics*(*length*,*length*) = *ics*(*deepness*,*profundity*) = 1

whereas:

ics(*flow*,*capacity*) = 0.

The similarity of the sets of attributes of complexTypes (*asim*) is therefore defined by the above maximum sum divided by the greatest of the cardinalities of the sets of attributes of the types compared.

In the case of *RiverType* and *CanalType* we have:

asim(*RiverType*,*CanalType*) = ¾ = 0.75

Finally, a boolean function is defined, namely B_t that, given two complexTypes, returns 0 if their least upper bound in the type hierarchy of Figure 1 is *AbstractFeatureType*, and returns 1 for any other case. For instance, in the case of *RiverType* and *CanalType*:

B_t(*RiverType*,*CanalType*) = 1

as their least upper bound is *MultiLineStringType*. This function essentially allows the similarity of elements whose types are derived from different GML complexTypes (such as, for instance, *MultiLineStringType* and *MultiPolygonType*) to be set to zero.

On the basis of the *ics*, *asim*, and B_t, we can evaluate the similarity (*GSim*) of GML elements. In essence, this is computed according to a weighted average between the *ics* of the element names and the *asim* of the related complexTypes, up to the normalization of the boolean function B_t. For instance, in the case of *River* and *Canal*, the following holds:

GSim(*River*,*Canal*) = (*ics* (*River*,*Canal*)*w +*asim*(*RiverType*,*CanalType*)(1-w)) * B_t(*RiverType*,*CanalType*)

In particular, if we assume w= ½, according to the previous results, we have:

$GSim(River,Canal) = ½ (0.72+0.75)*1 = 0.74$

With the same assumptions as above, we now consider the *ComWay* and *Canal* elements. In this case, as *River* has been replaced by *ComWay* (which is the least upper bound between *River* and *Canal*), the *ics* increases, whereas the *asim* decreases due to the presence of two additional attributes in *CanalType* with respect to *ComWayType*. In particular:

$GSim(ComWay,\text{Canal}) = (ics(ComWay,Canal)*w + asim(ComWayType,CanalType)(1-w))* B_t(ComWayType,CanalType) = ½ (0.85+0.50)*1 = 0.68$

since:

$ics(ComWay,Canal) = 2\ logp(ComWay)/(logp(ComWay) + logp(Canal)) = 0.85$
$asim(ComWayType,CanalType) = 2/4 = 0.50.$

Finally, consider navigable rivers (*NavRiver*) and non-navigable rivers (*NNavRiver*). Similarity decreases with respect to the previous cases, due to a "worst match" between the related sets of attributes (*asim*). In fact:

$GSim(NavRiver,NNavRiver) = (ics(NavRiver,NNavRiver)*w + asim(NavSegmentType, NNavSegmentType) (1-w)) * B_t(NavSegmentType, NNavSegmentType) = ½ (0.80+0.33)*1 = 0.56$

where:

$ics(NavRiver,NNavRiver) = 2\ logp(River)/(logp(NavRiver) + logp(NNavRiver)) = 0.80$
$asim(NavSegmentType, NNavSegmentType) = 2/6 = 0.33.$

As a final remark, it should be remembered that in general sequences of elements which are not parts, such as *Kind* and *Country*, must also be compared according to the algorithm mentioned above (this issue has been omitted in this paper due to lack of space).

5 Conclusion

In this paper we proposed a method to evaluate the semantic similarity of GML elements (concepts). Considering the importance of the *Is-in* relationship in the geographic context, we focused on GML elements organized according to *Part-of* (*meronymic*) hierarchies and proposed the application of our method to *Part-of* taxonomies, due to the semantics of the meronymic relationship within the geographic context, referred to as "place-area". This semantics essentially refers to parts similar to and inseparable from the whole. A further contribution of this paper is the modeling of the *Part-of* hierarchy in GML. In particular, from the perspective of applying the information content approach to *Part-of* hierarchies, we proposed a method to represent and distinguish the concepts involved in the meronymic relationship within the element hierarchy.

References

1. M.A.Rodriguez, M.J.Egenhofer "Comparing Geospatial Entity Classes: an Asymmetric and Content-Dependent Similarity Measure" International Journal of Geographical Information Science 18(3), pp. 229-256, 2004.
2. [ISO04] ISO/TC 211/WG 4/PT 19136 N 005r3. Geographic Markup Language (GML) 02-07-2004.
3. M.E.Winston, R.Chaffin, D.Hermann. "A taxonomy of part-whole relations" Cognitive Science, 11, 417--444, 1987.
4. P.Resnik "Semantic simiratity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language" Journal of Artificial Intelligence Research, Vol.11, pp. 95-130, 1999.
5. M.A.Rodriguez, M.J.Egenhofer "Determining Semantic Similarity among Entity Classes from Different Ontologies" IEEE Transactions on Knowledge and Data Engineering, Vol.15, n.2, pp. 442-456, 2003.
6. D.Lin "An Information-Theoretic Definition of Similarity" Proceed. 15th Intern. Conference on Machine Learning, ICML'98, Madison, WI, pp. 296-304, 1998.
7. P.Resnik "Using information content to evaluate semantic similarity in a taxonomy" Proceed. IJCAI 1995.
8. V.C.Storey "Understanding Semantic Relationships" Very Large Data Bases Journal, Vol.2, pp. 455-488, 1993.
9. L. Feng, E. Chang, T. Dillon "A Semantic Network-Based Design Methodology for XML Documents" ACM Transaction on Information Systems, Vol.20, No. 4, pp. 390-421, 2002.
10. P. Bouquet, A. Donà, L. Serafini, S. Zanobini "ConTeXtualized local ontology specification via CTXML" Proceedings of thee AAAI Workshop on Meaning Negotiation, Edmonton (Alberta), Canada, July 28, 2002.
11. J.H.Lee, M.H.Kim, Y.J.Lee "Information retrieval based on conceptual distance in is-a hierarchies" Journal of Documentation, Vol.49, n.2, pp.188-207, 1989.
12. R.Rada, H.Mili, E.Bicknell, M.Blettner "Development and application of a metric on semantic nets" IEEE Transaction on Systems, Man, and Cybernetics, Vol.19, n.1, pp. 17-30, 1989.
13. M.Sussna "Word sense disambiguation for free-text indexing using a massive semantic network" Proceed. Second Intern. Conference on Information and Knowledge Management (CIKM 93), Arlington, Virginia, 1993.
14. R.Richardson, A.F.Smeaton, J.Murphy "Using WordNet as a knowledge base for measuring semantic similarity between words. Working paper CA-1294, Dublin City University, School of Computer Applications, Dublin, Ireland, 1994.
15. V.Morocho, L.Perez-Vidal, F.Saltor "Semantic integration on spatial databases SIT-SD prototype" Proceed. Of VIII Jornadas de Ingenieria del Software y Bases de Datos, Alicante, Spain, pp. 603-612, Nov. 2003.
16. A.G.Maguitman, F.Menczer, H.Roinestad, A.Vespignani "Algorithmic detection of semantic similarity", Intern. World Wide Web Conference Commettee (IW3C2), WWW'05, Chiba, Japan, pp. 107-116, May 10-14, 2005.
17. S.Ross. "A First Course in Probability. Macmillan", 1976.
18. F.Menczer "Combining link and content analysis to estimate semantic similarity" Intern. World Wide Web Conference, WWW'04, New York, USA, May 17-22, pp. 452-453, 2004
19. W.B.Frakes, R.Baeza-Yates Ed.s "Information Retrieval, Data Structure and Algorithms" Prentice Hall, 1992 .
20. S.Chien, N.Immorlica "Semantic Similarity between Search Engine Queries using Temporal Correlation" Intern. World Wide Web Conference Commettee (IW3C2), WWW'05, Chiba, Japan, pp. 2-11, May 10-14, 2005.

21. A.Formica. “Similarity of XML-Schema Elements supported by Domain Ontologies”, International Journal of Software Engineering and Knowledge Engineering (IJSEKE), 15(1), 117-130, 2005.
22. WordNet 2.0. http://www.cogsci.princeton.edu/cgi-bin/webwn.
23. A.Formica, M.Missikoff. “Concept Similarity in SymOntos: an Enterprise Ontology Management Tool” The Computer Journal, 45(6), 583--594, 2002.
24. Z.Galil. “Efficient algorithms for finding maximum matching in graphs” ACM Computing Surveys, 18, 23--38, 1986.

Measuring Semantic Differences Between Conceptualisations:

The Portuguese Water Bodies Case – Does Education Matter?

Paulo Pires[1], Marco Painho[2], and Werner Kuhn[3]

[1] Superior Institute of Statistics and Information Management,
New University of Lisbon, Portugal
ppires@isegi.unl.pt
[2] Superior Institute of Statistics and Information Management,
New University of Lisbon, Portugal
painho@isegi.unl.pt
[3] Institute for Geoinformatics, University of Münster, Germany
kuhn@ifgi.uni-muenster.de

Abstract. In 2003 Pires [9] published a study that compared the results from the study performed by David Mark and Barry Smith[1] with a similar study applied to Portuguese subjects. The paper concluded that the methodology of Mark and Smith, to establish ontologies from surveys of how users apply terminology, is applicable to identify conceptualization differences in GIS applications.

This paper is an extension of that work, presenting the results of the study in terms of university background. In response to series of differently phrased elicitations, 533 subjects (university students from several parts of Portugal and several academic disciplines) were asked to give examples of geographical categories. By this we statistically counted the most mentioned terms and related these to the university background.

The results were analysed in order to test the hypothesis: Students from different backgrounds have different conceptualizations of geographical categories due to their scholarly background. Our analysis refutes this hypothesis: students present the same examples for the presented categories and their disciplinary backgrounds cannot be shown to have an influence on the category choices.

Ontology has been conceived as a branch of metaphysics that studies the theory of objects and their relationships [3].

In this paper we aim to explore the relation Ontology/Geographical Information Systems from the cognition perspective. We raise the question; does scholarly background influence geographic categorization? In order to answer this question we used a survey that studied a specific set of geographic concepts, water bodies. The main reason behind the choice of these specific entities is that Portugal is a country exposed to the Atlantic and where water has been considered as an important element since the time of the Discoverers.

The survey is based on a similar approach taken in other parts of the world, such as England, Finland and others [11].

Keywords: Cognition, Geographical Information Systems, Semantics, Geographic Categories.

[1] www.tandf.co.uk.journals. *IEEE Transactions on Knowledge and Data Engineering* , 15 (2): 442-456, 2003.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1020 – 1026, 2005.
© Springer-Verlag Berlin Heidelberg 2005

1 Ontology, the Meaning

"Specification of a conceptualisation" [4] is a very common answer to the question: what is an ontology? In philosophy, ontology refers to the subject of existence. When applied to information systems it refers to the description of concepts and relation of entities to a certain reality/domain.

The study by Mark and Turk [7] is a clear example of the importance Ontologies have in GIS. They studied the Yinjibarndi (a community of Australian Indigenous) terms for topographic features. When compared to English language terms to describe the Australian landscape they realised that there are some differences. For example, for the Yindjibarndi a river is a place that has a spirit, what they call a yinda. However a yinda, as long as it is permanent may be big, small, wide or narrow.

If a Geographical Information System is to be constructed in order to support the native title land claim, these differences need to be taken in to account, so that possible management arrangements (e.g. a national park) between the State Government and the Yindjibarndi people will be successful.

The first goal of our study is to contribute to the understanding of how Portuguese subjects from different backgrounds use geographical categories.

Ontologies then, appear as a means to help solve what may be called the "Tower of Babel" problem in GIS applications. When data from different sources are put together, there are problems of terminological and conceptual incompatibilities. Solving them requires an understanding of how different information communities use terms to refer to concepts.

In this context an ontology is a description of entities and their relationships in a determined universe/domain. In other words, as Gruber [4] defends, an ontology can be seen as "a specification of a conceptualisation".

2 Methodology – A Survey Applied to University Students

Several possible approaches can be used in order to understand what perceptions people have of a specific matter. One of the most common methods applied is the use of surveys.

For logistical reasons university students are often used for such studies, in spite of having certain characteristics that can cause some deviations in the analysis (being above the average in some critical variables in relation to the general population). In this situation, although they cannot be considered representative of all the Portuguese population, we have opted for using them as a sample for our consideration, for practical reasons.

The survey was applied to 574 students from several universities from various regions (centre and south) of Portugal: Coimbra, Lisboa, Évora and Faro (see figure 1) paying attention to the choice of an equal percentage (where possible) from each degree. This choice guaranteed the quality of the data for our purposes, regarding the available resources. In the first approach of this study we used 196 of those surveys. In this paper we present the results for all of the surveys (533), taking in consideration that the remaining ones cannot be used.

The fieldwork was performed between May 1st and July 15th 2003.

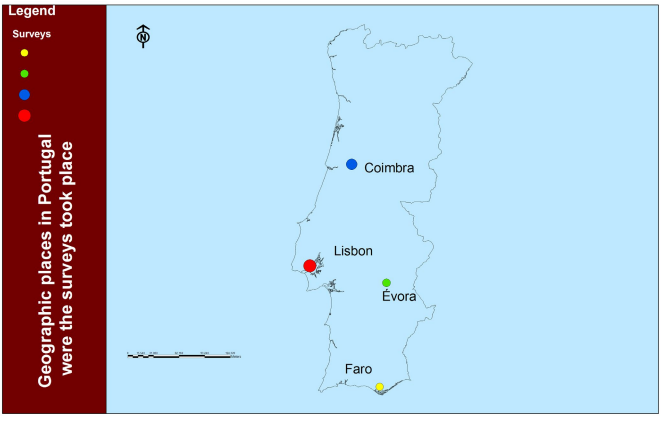


Fig. 1. Geographic places in Portugal were the surveys took place

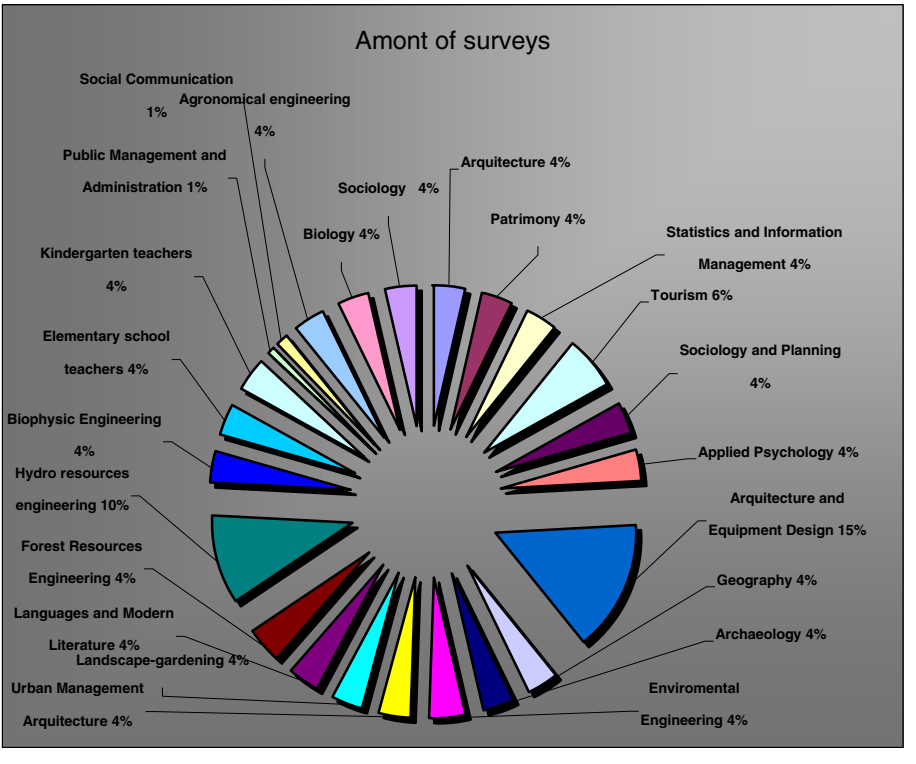


Fig. 2. Scholar background of the students

The survey was applied to different degrees, so that, we would have a significant sample; and so that we could have the conceptualisations of subjects with different backgrounds as well.

In figure 2 we are able to see the diversity of degrees (we were not able to analyse all 23 degrees because on 3 of them we had less then 5 surveys):

In response to a series of differently category titles, subjects were asked to give examples of geographical categories. By this we were able to statistically count the most mentioned terms.

For more detailed information on the methodological aspects, see [10].

3 Results – Scholarly Background and Geographical Categories

Students were asked to give examples of five categories: Natural earth formations; something that can be portrayed on a map; types of geographic objects; geographic feature; something geographic; geographic concepts. For each one of these categories they could fill in 5 examples. Most of the students had some difficulties in giving five examples, for each category. The analysis was thus concentrate only on terms mentioned at least 10% of the subjects for the listed five phrasings. However, as the dispersion of answers at lower levels was significant, we decided to accept as criterion, instead, to study only terms that were listed by at least 2% of the subjects for at least one of the five categories.

We discuss the results in the following sections.

3.1 Natural Earth Formation

The degrees of Architecture and Equipment Design, and Landscape Architecture were the ones that presented a bigger dispersion in the results, but none the less, most students tend to agree when it comes to what is a natural earth formation, the most common examples were *mountain*, *volcano* and water related entities such as *river* and *water*.

Only in the degree of Elementary school teachers, some students gave examples (with a high level of dispersion) of man made things, such as *road*.

For most students a natural earth formation was understood as something that was done by nature alone with no interference from man.

3.2 Something That Could Be Portrayed on a Map

In this category there is no doubt that most students mention *river* as being something portrayed on a map. The remaining students mention a water related example as first choice example for this category.

The students from geography were the ones that were able to give more examples in this category, thus the large dispersion of results. But all examples given by them were essentially more physical geographical characteristics rather than human ones (this was the case for all students of all degrees).

3.3 A Kind of Geographic Object

In this category the Portuguese student's results show less tendency to converge. The items indicated by the subjects, show similar tendency to indicate many examples of small, portable items. Map is the item most commonly mentioned. Several other top expressions include e.g., *compass, GPS, astrolabe*. This can be explained by the inclusion of the word object in the category witch would refer more to man made rather than natural things.

Geography, Information management, Biophysics Engineering and Public Administration Management students are the only ones to present different examples for this category (with a representative percentage). Geographers give examples of *population and Economic activities* with 5%. Where as Information Managers mention *point, line* and *map* 7,5% of the times. On the other hand, Engineers mention the example *"Serra"* (kind of mountain) in 5% of their answers. Finally Public Administration Management students also mention a physical natural formation, but in this case *river*, followed by a diversity of examples, to other degrees such as *plantform* or *farm*.

3.4 A Geographic Feature

In this category we have to take in consideration the fact that in Portuguese, the word "feature" is used in a more restricted sense, for geographical purposes, then in the English language. The Portuguese translated word ("característica"), is associated to the sense of "characteristics" or "properties" of an object/subject. That explains why the Portuguese answers are more centred on conceptual geographic items than in concrete geographic features.

There is no doubt that Portuguese students see geographic features/characteristics associated with cartographic concepts, because the most mentioned items were *latitude, longitude* and *altitude*. Some (mostly from architecture) also mentioned demographic concepts, but in a much lower frequency.

As expected we found examples of similar items mentioned as in the category "a geographic concept" and we did, this might mean that there is a thin line between distinguishing what is concept and characteristic. Students did not associate at all characteristics like *small, long, salty*, that is to say, generic characteristics, we assume they had interpreted geographic characteristic as geographic terminology.

3.5 Something Geographic

It is possible to say that the results from this category show the same tendency as the others, they clearly reinforce the idea that the results are very similar within scholar background. Portuguese students mentioned *map, river,* and *mountain* in the top places as something geographic.

This category leads to a more generalist approach to the geographic contents; all students mentioned the same items that they had already mentioned in the other categories. This confirms the fact that it is the most generalist category of all. People tend to indicate here the same examples that are also popular in other categories, essentially those relating to physical or artificial features (*river, mountain, city, lake*) or representation methods (*map*), but not specific items to this category, that is particularly evident since no new reference to new items was mentioned in this category.

3.6 A Geographic Concept

The Portuguese students had difficulty in answering this category. We observed here the highest number of missing values and this is reinforced by the dispersion of responses in all degrees.

Simultaneously, this is the category in which there are less common items in the selected top responses from all degrees. For example, arquitecture students mention altitude and longitude. As for the psychology students, they mention death rate and birth rate, and geography students that mention location.

In this category demographics and cartographic concepts are the most common e.g. latitude, longitude, altitude, population, population density and birth.

As mentioned above, we can also read from the results that subjects had a difficulty in deciding what a geographic concept is, assuming this category as a wider and more generic one.

4 Conclusions

Ontology, since Aristotle has been conceived as a branch of metaphysics that studies the theory of objects and their ties. Today, ontologies can improve the creation of geographical information in order to support human activities.

This paper aimed to contribute to the understanding on how different backgrounds influence the conceptualisation of geographic categories in Portugal.

We started by posing the hypothesis: Students from different backgrounds have different conceptualizations of geographical categories due to their scholar background.

Based on a specific section of the survey we wanted to understand if subjects from different backgrounds, in response to a series of differently phrased elicitations, would give the same examples of geographical categories.

All sets of students gave similar answers, mostly of a physical nature, like *mountain* or *river*. This can be explained by the fact that we are dealing with a universal concept: "geography". This means that social, economical or cultural characteristics of the population have little influence, the results will be similar.

This is a tendency much clearly marked, not only eventually for a language question, but mainly for the conditions in under which the survey was applied. A subject will tend to extremate and differentiate the items referred in each question as they move along the categories. It is accepted that the answers could be different if the questions were placed isolated and not sequentially. In figure 3, we are able to see the most relevant answers students gave:

Category	**Natural Earth Formation**	**Something that could be portrayed on a map**	**A kind of geographic object**	**A geographic feature**	**Something Geographic**	**Geographic concept**
Answers	Mountain (100%)	River (100%)	Map (99%)	Longitude (99%)	River (99%)	Longitude (99%)
	River (100%)	Ocean (99%)	Compass (99%)	Latitude (99%)	Mountain (99%)	Latitude (99%)

Fig. 3. Most mentioned answers for the six categories

Meaning that for example in the category "Something Geographic", 99% of the students gave *river* and *mountain* as examples under this category.

Future analysis with this data set includes the exploration of differences in answering according to geographical origin of the subjects.

Despite these results, this approach has some limitations and still presents considerable work to be done and new fields of study to explore. Something to be investigated is if not even the geographical differences (e.g. between the cost side and the interior of Portugal) seem to be a determining factor for different answers.

References

1. Battig, W. F. and W. E. Montague (1968), "Category norms for verbal items in 56 categories a replication and extension of the Connecticut Norms", *Journal of Experimental Psychology Monograph*, 80, Part 2, pp. 1-46
2. Butchvarov, P. (1995), "Category", in Kim, J. and E. Sosa (Eds.), *A Companion to Metaphysics*, Cambridge / Oxford: Basil Blackwell, pp. 75-79
3. Chandrasekaran, B. and J. Josephson (1999), "What are ontologies, and why do we need them?", *IEEE Intelligent Systems*, January/February 1999
4. Gruber, T. (2001), "What is an Ontology", available on www-ksl.Stanford.edu/kst/what-is-an-ontology.html
5. Guarino, N. (1998), "Formal Ontology and Information Systems", in Guarino N. (Ed.) *Formal Ontology and Information Systems*, Amsterdam: IOS Press, pp. 3-15
6. Kuhn, W. (2001), "Ontologies in Support of Activities in Geographic Space", *International Journal of Geographical Information Science*, vol. 15, nº 7, pp. 613-631
7. Mark, D. and A. Turk (2003), "Landscape Categories in Yindjibarndi: Ontology, Environment, and Language", in Kuhn, W., Sabine, T., and M. Worboys (Eds.), *Proceedings of the International Conference, COSIT 2003 International*, Berlin: Springer Verlag, pp. 28-45
8. Möltgen, J. and W. Kuhn (2000), "Task analysis in transportation planning for user interface metaphor design", Paper presented at the *3rd AGILE conference on GIS* in Helsinki, May, 25-27th
9. Pires, P. and Brox, Cristoph, (2003), "Measuring Semantic Differences between Experts' and Non-experts conceptualisations", Semantic conference in Mexico City, Mexico
10. Pires, P. (2005), Geospatial conceptualisation: A Cross-Cultural Analysis on Portuguese and American Geographical Categorisations, Journal of Data Semantics (LNCS subline, Springer), Special Issue on Semantic-Based Geographical Information Systems
11. Smith, B. and D. Mark (2001), "Geographical categories: an ontological investigation", *International Journal of Geographical Information Science*, vol. 15, nº 7, pp. 591-612

Spatio-temporal Schema Integration with Validation: A Practical Approach*

A. Sotnykova[1], N. Cullot[2], and C. Vangenot[1]

[1] École Polytechnique Fédérale de Lausanne, Database Laboratory, CH-1015 Lausanne, Switzerland
{Anastasiya.Sotnykova, Christelle.Vangenot}@epfl.ch
[2] Université de Bourgogne, Laboratoire LE2I, F-21078 Dijon, Cedex
Nadine.Cullot@u-bourgogne.fr

Abstract. We propose to enhance a schema integration process with a validation phase employing logic-based data models. In our methodology, we validate the source schemas against the data model; the inter-schema mappings are validated against the semantics of the data model and the syntax of the correspondence language. In this paper, we focus on how to employ a reasoning engine to validate spatio-temporal schemas and describe where the reasoning engine is plugged into our integration methodology. The validation phase distinguishes our integration methodology from other approaches. We shift the emphasis on automation from the a priori discovery to the a posteriori checking of the inter-schema mappings. By doing so, we take advantage of the expressive power of the common data model in the source schema description and inter-schema mapping definition.

1 Integration Approach

Database integration has been and continues to be the focus of many research efforts. The integration task involving the spatial and temporal domains becomes even more complex bearing a weak number of formal approaches reported in the literature. In this paper, we describe a hybrid approach exploiting advantages of two formalisms: a spatio-temporal conceptual model and an expressive description logic. The relationships between database conceptual models, in particular the Entity-Relationship (ER) model, and Description Logics (DLs) are addressed in [1,2], where it is shown that DLs are powerful enough to capture the domain semantics represented by ER based models. In this paper, we present a translation of MADS conceptual data model into a DL formalism. MADS [3] is an object+relationship conceptual data model featuring structural, spatial, temporal, and multiple perceptions dimensions. In our previous paper [4], we presented our integration methodology where MADS is the common data model

* This work is supported, in the framework of the EPFL Center for Global Computing, by the Swiss National Funding Agency OFES as part of the European projects KnowledgeWeb (FP6-507482) and DIP (FP6-507483).

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1027–1036, 2005.
© Springer-Verlag Berlin Heidelberg 2005

for the source and resulting integrated spatio-temporal database schemas. The methodology consists of four main phases where *1)* source schemas are translated into MADS data model; *2)* the inter-schema mappings are expressed in the MADS correspondence language; *3)* based on the set of inter-schema mappings and integrity constraints, the structural solutions for related parts of the schemas are proposed to the designer; and *4)* finally, the integrated schema is composed. Theoretical concepts underlying these phases include syntax and semantics for the inter-schema mapping languages and a predefined set of structural patterns for different set relationships between source schema populations. The integration process is not automatic: the source schemas are translated into the MADS model by the schema designer, the inter-schema mappings and the integrity constraints are asserted manually. Thus, our methodology requires a verification mechanism for the totality of the expressions stated manually.

We now propose an enhanced integration approach where every manual phase is supported by a validation operation ensuring data model and language compliance. For the phase *1)*, we aim to verify the validity of the expressions of the source schemas in the MADS model. The MADS data model itself and the source schemas with integrity constraints are translated into a DL based language and the satisfiability of the resulting translated DL model is validated. In phase *2)*, inter-schema mappings are added to the DL models in terms of elements of the source schemas and mapping operators [5] thus ensuring that the mappings are data model compliant and there are no contradictory mappings. Several possible ways to construct the integrated schema are then presented to the schema designer. During the phase *3)* these putative structural solutions are checked for compatibility with the integrity constraints imposed on the schema elements that are to be integrated. A successful satisfiability check signifies that the populations of the related schema elements can be merged; in case of failure, only the structural solutions that model the populations separately are applicable.The final decision on the choice of the structural solutions is up to the designer and/or a domain expert. By composing our integration method of the phases described above, we assure that the schema designer has sufficient knowledge about the compatibility of the source schemas and possible structural solutions for the final integrated schema.

The examples that we will use throughout the paper are two database schemas (Fig. 1.1 and 1.2) developed by two tourist offices describing the same geographical area, the city of Paris[1]. Schema T_1 models a cadastral division of the city, where the city is decomposed in several city boroughs and specifies where some of the urban services available for tourists, e.g., public transport stops available in each city borough. The object type TouristPlace is a spatio-temporal object type. Its spatial extension is of type simple area and its temporal one of type interval. The subtypes of TouristPlace inherit the spatial and temporal properties of their ancestor. The schema T_2 has a different focus. Tourist attractions that compose the population of the TouristSite object type are those accessible by public transport. The TouristSite object type has a spatial extension of type

[1] complete figures can be found in [4].

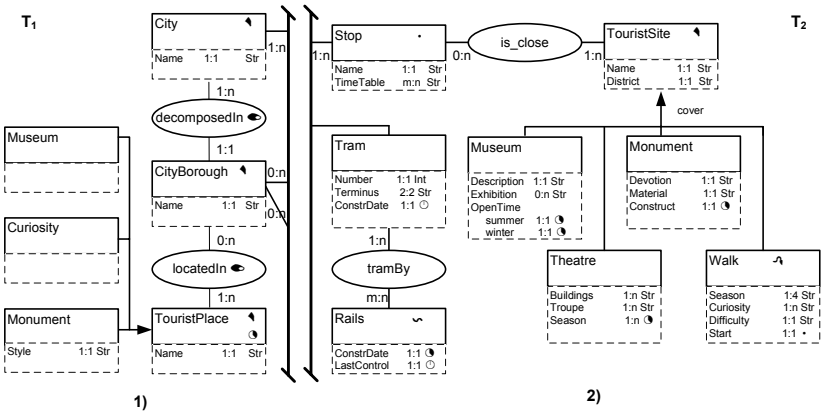


Fig. 1. Fragments of schema T_1 and schema T_2 both modeling a city for tourists

simple area and Stop of type point. Given those properties and the cardinality of the is_close relationship, the population of TouristSite is limited to the tourist attractions located not further than at a certain distance from a transport stop. The subtypes of TouristSite inherit its spatiality except for the Walk subtype whose spatial extension is redefined to the oriented line type.

2 Validation: Practice

Theoretically, we were looking for the description logic(s) best suited for modeling spatial and temporal data. As discussed in [4], to provide validation to the theoretical foundations of our integration approach, we exploited the $\mathcal{ALCRP}(\mathcal{S}_2 \oplus \mathcal{T})$ expressive power to describe the source spatio-temporal schemas and inter-schema mappings. The $\mathcal{ALCRP}(\mathcal{D})$ DL proposed in [6], extends $\mathcal{ALC}(\mathcal{D})$ to build complex roles based on a role-forming predicate operator [7]. In particular, an appropriate concrete domain $\mathcal{S}_2$ is defined for polygons using $\mathcal{RCC}_8$ relations as basic predicates of concrete domain. For temporal aspects, the concrete domain $\mathcal{T}$ is a set of time intervals and Allen relationships are used as basic predicates describing the relationships between intervals.

In practice, as we want to use an existing reasoner, we are limited by the $\mathcal{SHOIN}(\mathcal{D})$ [2] description logic which is the base for the OWL ontology language. For this logic we can use Protégé [8] graphical tool, and RACER [9] reasoning system. The $\mathcal{SHOIN}(\mathcal{D})$ logic does not treat either spatial, or temporal concrete domains. Next, we will show how we implement the MADS model in this logic aiming at emulating spatial and temporal semantics for describing spatio-temporal information.

2.1 Data Model Definition in OWL

In this section we present the translation of the structural, spatial, and temporal dimensions of the MADS data model in the OWL (OWL-DL) language. Due

to the space limitation we will not detail MADS constrained relationships and multiple perceptions. Below, we refer to the MADS model expressed in OWL as MADS-OWL.

Structural Dimension. An OWL model is composed of classes, properties, and instances. Restrictions on properties and classes are the integral part of an OWL model. In MADS, a schema is composed of object and relationship types, and attributes.

Object types definition. We define a MADS object type in OWL by `<owl:Class>` constructor. A MADS object type can have several subtypes and supertypes, allowing for the multi-inheritance which is an important feature of the MADS model. In OWL, classes may also have several subtypes and supertypes. OWL classes by default, share their instances; in MADS, object types have explicit extensions. To restrict an OWL class, for example, the TouristPlace class to have only predefined set of instances we use the `<owl:unionOf>` axiom over its subclasses:

```
<owl:Class rdf:ID="TouristPlace">
    <rdfs:subClassOf>
    <owl:Class>
    <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:ID="Museum"/>
        <owl:Class rdf:ID="Curiosity"/>
        <owl:Class rdf:ID="Monument"/>
    </owl:unionOf>
    </owl:Class>
    </rdfs:subClassOf>
</owl:Class>
```

With the above described OWL features, we are able define object types, the generalization/specialization links with disjoint and cover axioms, and multi-inheritance as in the MADS model.

Relationship types definition. A MADS relationship is a link between two object types. An OWL property is a binary relationship that connects two OWL concepts, called domain and range of the property. Linked concepts can be either classes, or a class and a datatype thus defining two types of properties: object properties linking individuals to individuals; and datatype properties linking individuals to data values. Unless otherwise constrained, OWL properties are multivalued. To represent MADS relationships we use OWL object properties, for example, MADS relationship tramBy from Fig. 1.2 is defined as follows:

```
<owl:ObjectProperty rdf:ID=tramBy">
    <rdfs:domain rdf:ID="Tram"/>
    <rdfs:range rdf:ID="Rails"/>
</owl:ObjectProperty>
```

Like for the object types, in MADS we can define the generalization/specialization link between relationship types. OWL also support hierarchies of properties which can be specified with `<owl:subPropertyOf>` axiom.

Attributes definition. MADS attributes can either be complex or simple, monovalued or multivalued, mandatory or optional. A MADS attribute is translated to OWL by one of the two property types depending on the type of the attribute. To represent MADS simple attributes, like the Description attribute of the object type Museum from Fig. 1.2, we create an OWL property Description, with the domain restricted to the class Museum, and the range of the datatype string:

```
<owl:DatatypeProperty rdf:ID="description">
    <rdfs:domain rdf:ID="Museum"/>
    <rdfs:range rdf:ID="http://www.w3.org/2001/XMLSchema string"/>
</owl:DatatypeProperty>
```

Defining a complex attribute in OWL requires additional classes to be created. For instance, Museum has a complex attribute OpenTime with two component attributes summer and winter, Fig. 1.2. To translate MADS attribute OpenTime to OWL property OpenTime, we first create a class MuseumOpenTime, and then two object properties summer and winter with the domain restricted to this class. Then the object property OpenTime with domain Museum and range MuseumOpenTime can be created. Each value of the property openTime would be an instance of a class MuseumOpenTime with two values for winter and summer properties. We summarize all the structural elements that we use in two modeling approaches - MADS and OWL, in Table 1.

Table 1. Structural Dimension - MADS vs OWL

MADS concepts	**OWL constructors / restrictions**
Object type	`owl:Class`
IsA link	`owl:SubClassOf`
Covering axioms: cover, disjoint	`owl:UnionOf`,`owl:DisjointWith`
Relationship type	`owl:ObjectProperty` with defined `owl:range` `owl:domain`, and `owl:inversePropertyOf`
IsA link	`owl:subPropertyOf`
Role cardinalities	`owl:minCardinality`, `owl:maxCardinality` `owl:Cardinality`
Object simple attribute	`owl:datatypeProperty`
Object complex attribute	`owl:objectProperty`
Identifier attribute	`owl:functionalProperty`

Spatial Dimension. The MADS spatial dimension is defined by the hierarchy of spatial abstract data types (SADT) and the set of topological relationships. The SADTs are associated to object types, and attributes to add specific spatial features whereas topological features are added to relationship types to convey spatial constraints. Spatial dimension in MADS is orthogonal to the structural dimension. Similarly, modeling in OWL we aim at defining the spatial dimension in a way it could be freely added or removed as additional feature to structural elements. We define a hierarchy of OWL classes together with the restrictions on the topological properties that may hold between spatial classes. The user

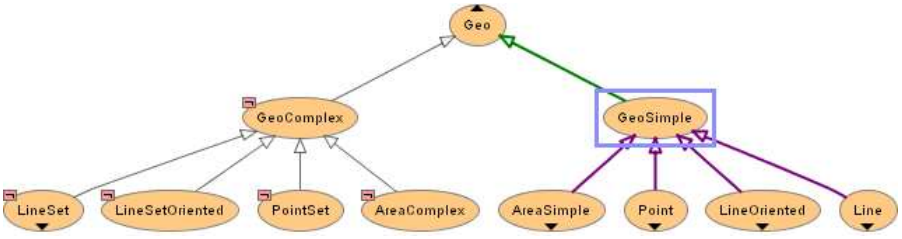


Fig. 2. OWL: the hierarchy of spatial classes

spatial classes are defined as subclasses of the members of the spatial hierarchy according the chosen spatial features. With the `<owl:subClassOf>` constructor, the restrictions of the superclass are inherited by its subclasses, preserving its spatial behavior and allowing controls on the topological relationships between spatial classes.

Contrary to as how the spatial features are defined in MADS, in MADS-OWL we only introduce the hasGeometry property as a synthetic property without adding all the semantics of its MADS counterpart. We make it intrinsic or defining property of spatial classes. We define the root spatial class Geo, and restrict the domain of the property hasGeometry to it; the range of this property we restrict to the union of classes GSimple and GComplex which are two disjoint classes populated with spatial instances. Then, we define the complete OWL spatial hierarchy extending Geo class, Fig. 2; and precising the values for the hasGeometry property for each spatial subclass.

Temporal Dimension. Temporal dimension is defined in a similar way to the spatial dimension. In compliance to the MADS temporal abstract type hierarchy we define the temporal MADS-OWL class hierarchy as shown in Fig. 3. The intrinsic property for temporal types is the hasTime property, meaning that the necessary and sufficient condition for a class to be classified as a temporal one, is the existence of the hasTime property. For the root temporal class Time, this

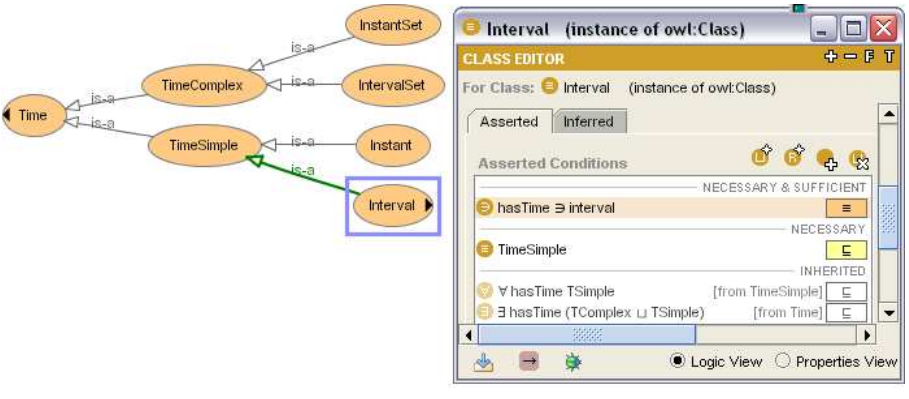


Fig. 3. OWL: Interval temporal class

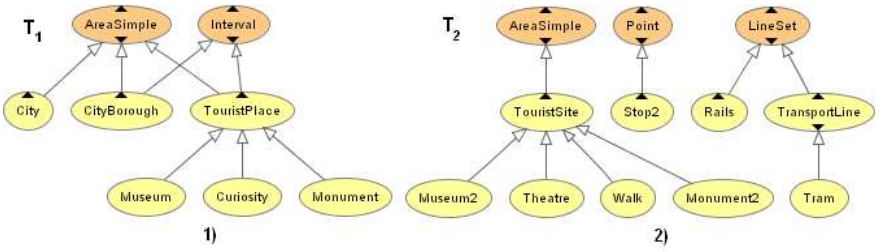


Fig. 4. Fragments of the schemas T_1 and T_2 in Protégé

property must exist, and the value of this property is restricted to one of the values from the TSimple or TComplex classes.

For the subclasses, we define additional axioms on the values of the property hasTime. For example, let us consider in details the temporal type Interval, its representation is the Protégé tool shown in Fig. 3. All the restrictions concern the hasTime property. From the root class Time, Interval inherits the most general restriction on the hasTime property. Then, from the TimeSimple class a more precise restriction is inherited. This restriction is compliant to the previous, more general one, and it restricts the range of the hasTime property only to the instances of the TSimple class. Finally, in the leaf class Interval, the restriction on the hasTime property is the most precise; the value of the property must be the interval instance of the TSimple class.

2.2 Schema Definition in OWL

With the MADS-OWL model, the designer can create his/her own models using the spatial and temporal dimensions of MADS-OWL. Fig. 4.1 depicts the classes that model schema T_1 in OWL. We do not show the relationships and attributes due to the complexity of the whole image. In MADS schema, the spatial object types TouristPlace, City, and CityBorough hold the simple area spatial semantics. Thus, in MADS-OWL we define them as subclasses of the AreaSimple spatial OWL class. The subclasses of TouristPlace, Museum, Curiosity, and Monument, inherit the spatial definitions of their superclass. As defined in MADS-OWL model, all the spatial classes are disjoint, i.e., a user-defined class cannot be a subclass of two spatial classes. Thus, all the subclasses of AreaSimple are disjoint with subclasses of Point. A MADS spatio-temporal object type is defined in MADS-OWL as a subclass of one of the temporal and one of the spatial classes. We define the TouristPlace object type from Fig. 1.1 as a subclass of the AreaSimple and Interval MADS-OWL types. Fig. 4.2 depicts the classes that model schema T_2 in OWL. Due to OWL unique name assumption, we have added index 2 to some classes in T_2, e.g., we name T_2.Museum as Museum2 in OWL.

To define a spatial attribute in MADS-OWL, we create an OWL property of type `<owl:objectProperty>` with the range restricted to a spatial class. For example, for the attribute Start of the object type Walk from Fig. 1.2, we define the OWL property start with the range restricted to the instances of the spatial class Point:

```
<owl:ObjectProperty rdf:ID="start">
    <rdfs:domain rdf:ID="Walk"/>
    <rdfs:range rdf:ID="Point"/>
</owl:ObjectProperty>
```

Properties in OWL are unidirectional, i.e., a MADS relationship is translated in OWL by two properties. Let us consider as an example the locatedIn relationship between object types TouristPlace and CityBorough from Fig. 1.1. This relationship holds topological inclusion semantics and therefore can link only spatial object types of specific spatial domains, e.g., a point can not include an area. In OWL, spatial property locatedIn is defined as a subproperty of the include_area topological property which belongs to the MADS-OWL model and restricts the domain and range of its subproperties by spatial classes:

```
<owl:ObjectProperty rdf:ID="locatedIn">
    <rdfs:subPropertyOf>
        <owl:ObjectProperty rdf:ID="include_area"/>
    </rdfs:subPropertyOf>
    <rdfs:domain rdf:ID="TouristPlace"/>
    <rdfs:range rdf:ID="CityBorough"/>
    <owl:inverseOf>
        <owl:ObjectProperty rdf:ID="locates"/>
    </owl:inverseOf>
</owl:ObjectProperty>
```

Note an inverse property locates that has CityBorough class as its domain and the TouristPlace class as its range. This property completes the definition of the MADS locatedIn relationship.

One of the assumptions that we make in our approach is the validity of the source schemas. Thus at this translation phase, ontology checks for each schema should not give any error. The subsumption check should not produce any new subclass/superclass relationships since all the classes within each schema are mutually disjoint. Consistency check will verify if there are contradictory restrictions, for example, cardinality constraints on relationships.

2.3 Inter-schema Mappings Definition in OWL

There are three types of the inter-schema mappings relating schemas that are to be integrated. Here we consider the first set of mappings, *Schema population Correspondences* that map related populations; we will use the same set of correspondences as we used in [4]:

(1) T_2.TouristSite $\cap$ T_1.TouristPlace;
(2) T_2.Museum $\subseteq$ T_1.Museum;
(3) T_2.Monument $\subseteq$ T_1.Monument;

In MADS, the *Schema population Correspondences* are stated between object types using the overlapping, inclusion, equality, and disjoint operators - $\{\cap, \subseteq, \equiv, \emptyset\}$. In OWL, classes are assumed to overlap, i.e., if it is not explicitly forbidden by the disjoint axiom, OWL classes can have common instances,

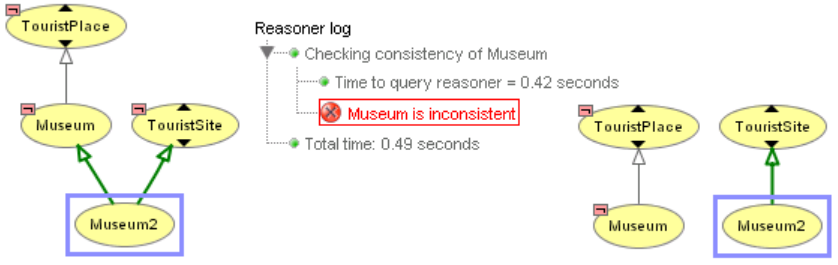


Fig. 5. Validation for the `<owl:subClassOf>` condition

which correspond to the overlapping relationship in MADS. Within each of our OWL schemas, we stated the `<owl:disjointWith>` axiom for all classes; but classes belonging to different schemas do not have this restriction. The inclusion in OWL is stated as the subclass restriction where subclass means necessary implication, i.e., if the T_2.Museum is stated as a subclass of T_1.Museum, then all the instances of the T_2.Museum are instances of T_1.Museum. This definition of subclasses in OWL corresponds exactly to the inclusion relationship definition in MADS. Thus, for the population correspondences (2) and (3) we have stated the corresponding `<owl:subClassOf>` axioms.

Let us now run the reasoner with an invalid condition which we have asserted intentionally: TouristPlace and TouristSite are disjoint. This condition invalidates the previous ones, as TouristPlace is the superclass of the Museum, and TouristSite is the superclass of Museum2, Fig. 4.1 and 4.2. The model asserts that with the disjoint superclasses some of their subclasses have common instances, as shown graphically in Fig. 5. If we check consistency for the Museum class, the reasoner finds it inconsistent. The inferred (consistent) model does not have the inclusion condition anymore.

For spatial and temporal mappings we use the MADS-OWL topological and synchronization properties respectively. For example, to state that for identical instances of the classes Museum2 and Museum their spatial properties are equal, we add the s_equal topological property to the Museum2 class. s_equal is the MADS-OWL property that expresses spatial equality; it is a symmetrical property and thus, it must have its range and domain of the same spatial type. If the designer states for example, that a museum is spatially equal to a bus stop, the reasoner will find the Museum class inconsistent. To ensure that the set of mappings is not contradictory, we added several constraints in Protégé axiom language to our model. For the spatial mappings for example, there is a constraint in MADS-OWL restricting two spatial classes from having any other spatial mappings if there is a s_disjoint (spatially disjoint) mapping between them. A similar constraint is added for MADS-OWL temporal dimension.

3 Conclusion

To represent MADS model definitions in OWL, we defined the MADS-OWL model that is imported as a reference ontology in the user model. Then, the

user defines spatial or temporal classes or properties by using the MADS-OWL ontology classes and properties as superior for his/her model elements. If the designer wants to rewrite the inherited values, his choice is checked for compatibility with parent values. If the designer defines a domain (or range) that is not a specialization of the domain (or range) of the parent property, this choice is rejected and an error message is displayed.

Our modeling approach insures that the model designer will correctly define spatial and temporal elements as well as the topological and synchronization properties. A model constructed with the help of the reference MADS-OWL model, can be checked for satisfiability considering MADS spatial and temporal semantics.

References

1. Calvanese, D., Lenzerini, M., Nardi, D.: Description logics for conceptual data modeling. In Chomicki, J., Saake, G., eds.: Logics for Databases and Information Systems. Kluwer Academic Publisher (1998) 229–263
2. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
3. Parent, C., Spaccapietra, S., Zimanvi, E., Donini, P., Plazanet, C., Vangenot, C.: Modeling spatial data in the MADS conceptual model. In: Proceedings of the International Symposium on Spatial Data Holding, Vancouver, Canada (1998)
4. Sotnykova, A., Vangenot, C., Cullot, N., Bennacer, N., Aufaure, M.A.: Semantic mappings in description logics for spatio-temporal database schema integration. Journal on Data Semantics III, Special Issue (2005) to appear.
5. Sotnykova, A., Monties, S., Spaccapietra, S.: Semantic integration in MADS conceptual model. In Bestougeff, H., Thuraisingham, B., eds.: Heterogeneous Information Exchange and Organizational Hubs. Kluwer (2002)
6. Haarslev, V., Lutz, C., Möller, R.: A description logic with concrete domains and a role-forming predicate operator. Journal of Logic and Computation **9** (1999)
7. Lutz, C.: Description logics with concrete domains—a survey. In: Advances in Modal Logics Volume 4, King's College Publications (2003)
8. (http://protege.stanford.edu/)
9. Haarslev, V., Möller, R.: Racer system description. In: Proc. of International Joint Conference on Automated Reasoning, IJCAR 2001, Springer-Verlag (2001) 701–705

Preserving Semantics When Transforming Conceptual Spatio-temporal Schemas

Esteban Zimányi and Mohammed Minout

Department of Computer & Network Engineering, CP 165/15,
Université Libre de Bruxelles,
50 av. F.D. Roosevelt, 1050 Brussels, Belgium
{ezimanyi, mminout}@ulb.ac.be

Abstract. Conceptual models provide powerful constructs for representing the semantics of real-world application domains. However, much of this semantics may be lost when translating a conceptual schema into a logical or a physical schema. The reason for this semantic loss is the limited expressive power of logical and physical models. In this paper we present a methodology that allows to transform conceptual schemas while preserving their semantics. This is realized with the help of integrity constraints that are automatically generated at the logical and physical levels. As a result, the semantics of an application domain is kept in the database as opposed to keeping it in the applications accessing the database. We present such a methodology using the MADS conceptual spatio-temporal model.

1 Introduction

During the conceptual modeling phase, the designer captures and describes the relevant actors and resources participating in the universe of discourse as well as the semantic links among them. The goal of this phase is to produce a conceptual schema composed of entity and relationship types as well as the associated constraints [3] such as cardinalities, topological, and synchronization constraints. Afterwards, the conceptual schema is translated into a logical and a physical schema that uses only the concepts supported by the target data management system (DBMS or GIS). Such translation induces a semantic loss because of the difference in expression power between conceptual models and the physical models of current data management platforms.

Therefore, integrity constraints are needed to ensure the semantic equivalence between the conceptual and the physical schemas. However, current DBMSs provide little support for *declarative* integrity constraints, besides very basic ones such as keys, referential integrity, or check constraints. For this reason, integrity constraints must be implemented using *triggers* [10]. A trigger is a named Event-Condition-Action rule that is automatically activated when a table is updated. Both SQL:2003, the most recent SQL standard, as well as major commercial DBMSs (such as Oracle and SQL Server) support triggers.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1037–1046, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Managing the constraints within the DBMS has the following advantages:

- Constraints are encoded once and for all in the database, instead of being encoded in each application accessing the database.
- Constraints are available to all (present and future) applications accessing the database, thus enforcing data quality.
- Constraints are encapsulated with the data, which from a software engineering perspective facilitates the overall lifecycle of the application.

In this paper we describe how to automatically generate a set of integrity constraints when translating a conceptual schema into a logical and a physical schema. We concentrate on spatial and temporal constraints. Section 2 briefly presents the MADS model. Section 3 explains with an example the translation of conceptual MADS schemas. Section 4 shows examples of temporal and spatial constraints generated while translating conceptual schemas. Finally, Section 5 concludes and points to directions for future research.

2 The MADS Model

MADS [6,7,8] is a conceptual spatio-temporal model. It includes four modeling dimensions: structural, spatial, temporal, and multi-representation. These modeling dimensions are *orthogonal*, meaning that spatial, temporal, and multi-representation features can be freely added to any structural construct of the schema: object and relationship types, attributes, methods, etc. Due to space limitations, we do not cover in this paper the multi-representation features of the MADS model. The other three dimensions are presented next.

Structural Dimension. MADS includes well-known features such as object and relationship types, attributes, and methods. Two types of relationships are provided: associations and multi-associations. In addition, relationships can be enhanced with semantic adornments such as aggregation, generation, and transition. Finally, MADS supports is-a links with rich multi-instantiation facilities.

Spatial and Temporal Dimensions. To describe the spatiality of real-world phenomena, MADS provides a set of spatial data types organized in a generalization hierarchy including generic (`Geo`), simple (e.g., `Point`, `Line`), and complex types (e.g., `PointSet`, `LineSet`). Similarly, there is a set of temporal data types, which are also organized in a hierarchy. Spatial and temporal data types have an associated set of methods to handle the instances of the type.

Spatiality and temporality can be associated both to types and to attributes. Spatial object or relationship types have an associated geometry. Temporal object or relationship types keep the lifecycle of their instances: objects or relationships are created, can be temporarily suspended, then reactivated, and finally disabled. The lifecycle is described by a particular attribute that can take one of four values: `scheduled`, `active`, `suspended`, or `disabled`. Further, spatial and temporal attributes can be attached to object or relationship types, independently of whether they are spatial/temporal or not.

Continuous fields are described with the concept of *varying attributes*, i.e., attributes whose values are defined by a function. Attributes may vary over space, time, and/or representation. For example, a space-varying attribute can be used for representing the depth of a lake while a space- and time-varying attribute can be used for representing moving objects.

Phenomena described by varying attributes can be *continuous* (like elevation), *stepwise* (like the type of crop in a cultivated area), or *discrete* (like mines in a minefield). Each type of function defines the kind of interpolation, if any, used to compute the value(s) of the attribute.

Constrained relationship types are relationship types conveying spatial and temporal constraints on the objects they link. Topological relationships define a spatial constraint [2] between the geometry of the related objects. For example, a topological relationship of type intersect may link the object types Flood and River, expressing that the geometries of a River and a related Flood intersect. MADS proposes a range of predefined topological relationships [4], such as disjoint, adjacent, intersects, etc. Similarly, synchronization relationships allow specifying constraints on the lifecycle of the participating objects. The semantics of the synchronization relationships is based on Allen's operators, e.g., before, equals, meets, etc., extended to complex temporal types.

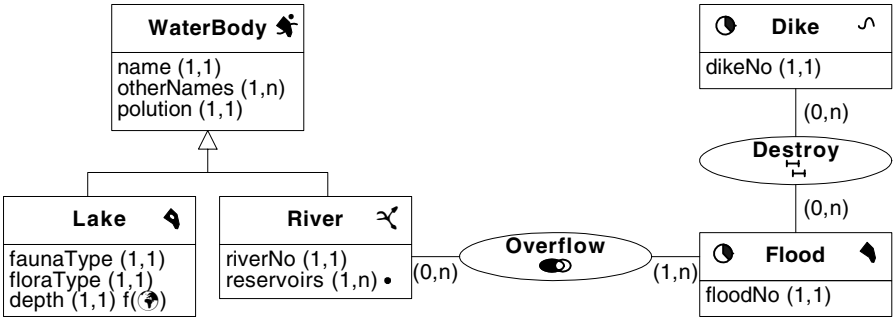


Fig. 1. Example of a MADS schema

Figure 1 shows the MADS schema for a river monitoring application. The spatial and temporal characteristics are visually represented by icons. The temporal icon on the left-hand side of the object/relationship types expresses that the lifecycle information is to be kept. Spatial icons are shown on the right-hand side. Spatial and temporal attributes are represented by the corresponding icon, as for the attribute reservoirs of River. Varying attributes are represented by the f( ) notation, as for the space-varying attribute depth of Lake. The Overflow relationship type is a topological semantics of type intersects. Similarly, the synchronization relationship Destroy is defined using a temporal intersects operator.

3 Translation of MADS Conceptual Schemas

3.1 Logical Schema

Since many concepts of the MADS model are not provided by current target platforms, we have developed a schema translator tool [5] that transforms a conceptual MADS schema into an equivalent logical schema that only employs the subset of MADS concepts provided by the target system. Figure 2 shows the object-relational logical schema obtained by translating the conceptual schema in Figure 1. We explain next some of these transformations.

Transformation of Relationships. A set of transformations rules allow to translate relationships according to their type and the cardinalities of the roles. For example, the many-to-many relationship Destroy is transformed by creating multivalued reference attributes floodRef and dikeRef in the object types Dike and Flood linked by the relationship.

Transformation of Spatial Object Types. For target systems supporting spatial attributes but not spatial object or relationship types (e.g., Oracle 10g), a spatial object or relationship type is transformed by creating a geometry attribute of the same spatial data type as the initial object type.

Another rule transforms specialized spatial data types (for example, surface for Lake in Figure 1) into the generic type Geo. This transformation is needed for models, such as Oracle 10g, having only one generic spatial type; the definition of the specific type (point, line, surface, ...) will be done at instance creation. This rule also generates an integrity constraint expressing that the spatial value must belong to the type initially specified in the conceptual schema.

Transformation of Varying Attributes. Since most systems do not support the concept of varying attribute, these are replaced by a complex multivalued attribute encoding its defining function. Thus, the attribute is composed of spatial, temporal, and representation extents (depending on how many dimensions the attribute varies) and a value. Figure 2 presents this transformation for the space-varying attribute depth. The first component attribute is a spatial attribute point and the second is the value of the attribute at this point.

Transformation of Temporal Object Types. Temporal object types, such as Flood and Dike in Figure 1, are transformed by creating a monovalued time-varying attribute, called lifecycle. A second transformation replaces this time-varying attribute by a multivalued complex attribute composed of a temporal data type (e.g., an interval) and a value. Since most systems only provide a domain of the instant type (for example the DATE type), a last rule transforms an interval into a complex attribute having the same name, the same cardinality, and whose component attributes (start and end) are of type instant (cf. Figure 2).

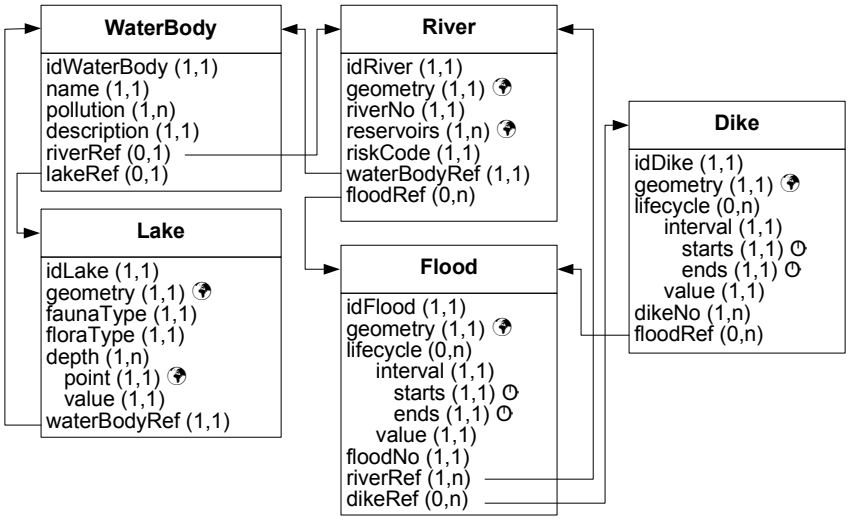


Fig. 2. Logical schema of Figure 1 for Oracle 10g (object-relational model)

3.2 Physical Schema for Oracle 10g

For each target DBMS and GIS there is a wrapper that rewrites the logical schema generated by the translator and produces a schema expressed in the language of the target system (e.g., SQL scripts). In the case of Oracle 10g object types generate object tables and multivalued domains generate either nested tables or VARRAY types. Links are replaced by using REF types that are logical pointers to row objects. For example, the dikeRef attribute in the object type Flood points to a nested table of references to Flood objects. An excerpt of the SQL script for the schema in Figure 2 is as follows (only the transformation of Flood is given):

```
create or replace type Did as object (id integer);
create or replace type DInterval as object (starts date, ends date);
create or replace type DLifecycleValue as object (interval DInterval,
   status varchar2(10));
create or replace type DLifecycle as table of DLifecycleValue;
create or replace type DRiverSetRef as table of ref DRiver;
create or replace type DDikeSetRef as table of ref DDike;
create or replace type DFlood as object (idFlood Did,
  geometry mdsys.sdo_geometry, lifecycle DLifecycle,
  floodNo integer, riverRef DRiverSetRef, dikeRef DDikeSetRef);
create table Flood of DFlood
   nested table lifecycle store as FloodLifecycleNT
   nested table riverRef store as FloodRiverRefNT
   nested table dikeRef store as FloodDikeRefNT;
alter table FloodLifecycleNT
```

```
    add constraint uniqueStarts unique (interval.starts));
alter table FloodLifecycleNT
    add constraint uniqueEnds unique (interval.ends));
alter table FloodLifecycleNT
    add constraint validInterval check (interval.starts < interval.ends);
alter table FloodLifecycleNT
    add constraint validStatus check (status in
      ('scheduled', 'active', 'suspended', 'disabled');
```

In the object-relational model constraints can only be defined on tables, not on types. For this reason four constraints are defined in the nested table FloodLifecycleNT for checking the validity of intervals and status values of the lifecycle. However, the disadvantage is that the same constraints should be defined for every nested table representing the lifecycle of a temporal type.

4 Temporal and Spatial Integrity Constraints

4.1 Temporal Constraints

Lifecycle of Temporal Types. The translation of lifecycles generates a set of temporal constraints. For example, one of them states that the intervals of the lifecycle must be disjoint, e.g., an instance cannot be active and suspended at the same time. Given the constraints on FloodLifecycleNT stated above, the following constraint at the logical level ensures this:

$$\forall f \in \mathsf{Flood}, \forall l_1 \in f.\mathsf{lifecycle}, \forall l_2 \in f.\mathsf{lifecycle}\ (\\ l_1.\mathsf{interval.starts} < l_2.\mathsf{interval.ends} \wedge l_2.\mathsf{interval.starts} < l_1.\mathsf{interval.ends} \Rightarrow \\ l_1.\mathsf{interval.starts} = l_2.\mathsf{interval.starts} \wedge l_1.\mathsf{interval.ends} = l_2.\mathsf{interval.ends}\ )$$

This constraint states that if intervals l_1 and l_2 of the lifecycle overlap, they must be the same interval. Another constraint states that once an instance has been disabled, it cannot change its status:

$$\forall f \in \mathsf{Flood}, \neg\ (\ \exists l_1 \in f.\mathsf{lifecycle}, \exists l_2 \in f.\mathsf{lifecycle}\ (\ l_1.\mathsf{status} = \text{'disabled'} \wedge \\ l_2.\mathsf{status} \neq \text{'disabled'} \wedge l_1.\mathsf{interval.starts} < l_2.\mathsf{interval.starts}\ )\ )$$

Each temporal constraint will be translated into one or several triggers in Oracle 10g. For the first constraint, one trigger will raise an error upon insertion of flood instances if two intervals of the lifecycle overlap.

```
create or replace trigger FloodLifecycleOverlappingIntervals
before insert on Flood for each row
begin
  if exists ( select * from table(:new.lifecycle) l1, table(:new.lifecycle) l2
    where l1.interval.starts < l2.interval.ends and l2.interval.starts < l1.interval.ends
    and l1.interval.starts <> l2.interval.starts )
  then
    raise_application_error(-20300,'Overlapping intervals in lifecycle')
  end if;
end
```

Similarly, a trigger for the second constraint will raise an error if the status is changed after being disabled.

```
create or replace trigger FloodLifecycleStatusChangeAfterDisabled
before insert on Flood for each row
begin
  if exists ( select * from table(:new.lifecycle) l1, table(:new.lifecycle) l2
    where l1.status = 'disabled' and l2.status != 'disabled'
    and l1.interval.starts < l2.interval.starts )
  then
    raise_application_error(-20301,'Change of status after being disabled')
  end if;
end
```

Synchronization Relationships. A synchronization relationship constrains the lifecycles of the linked object types. For example, in Figure 1 the synchronization relationship Destroy specifies that an instance of Flood can be related to an instance of Dike only if their lifecycles intersect.

As shown in Figure 2, the synchronization constraint is lost in the translation, only the underlying binary relationship is represented in the logical schema as a couple of multivalued reference attributes dikeRef and floodRef in tables Flood and Dike, respectively. A set of triggers is generated automatically for preserving the semantics of this relationship in the physical schema. One of them fires on insertions into Flood as follows:

```
create or replace trigger FloodDestroySynchronization
before insert on Flood for each row
begin
  if exists ( select * from table(:new.lifecycle) l1, table(:new.dikeRef) d,
    where not exists ( select * from table(d.column_value.lifecycle) l2,
      table(d.column_value.floodRef) f
      where f.column_value.idFlood=:new.idFlood
      and l1.interval.starts < l2.interval.ends and l2.interval.starts < l1.interval.ends
      and l1.status='active' and l2.status='active' )
  then
    raise_application_error(-20302,'Violation of Destroys synchronization relationship')
  end if;
end;
```

The purpose of this trigger is to rollback the insertion of a new instance of Flood if it is related to a dike d and it is not the case that d is related to the new instance of Flood and both instances are active during some time interval.

4.2 Spatial Constraints

Geometry of Spatial Types. The translation of spatial object and relationship types generates a geometry attribute keeping their spatiality. However, since Oracle only provides a generic spatial type, a constraint must enforce that the values of the attribute correspond to the spatial type defined in the conceptual

schema. For example, the following constraint ensures that the geometries of lakes are of type multiline or multicurve (type 6 in Oracle).

```
alter table Lake
  add constraint validGeometryType check (geometry.get_gtype() = 6);
```

Spatial Attributes. A set of integrity constraints is generated when translating spatial attributes. For example, one constraint verifies that each value of the attribute reservoirs of River is of spatial type point (type 2 in Oracle). Since reservoirs are stored in a VARRAY, the following trigger is needed.

```
create or replace trigger RiverReservoirsPointType
before insert on River for each row
begin
  if exists ( select * from table(:new.reservoirs) r where r.get_gtype() != 2 )
  then
    raise_application_error(-20401,'Reservoirs must be of spatial type point')
  end if;
end;
```

Another constraint verifies that the spatiality of reservoirs is inside the spatiality of River. This constraint it expressed at the logical level as follows:

$$\forall r_1 \in \text{River}, \forall r_2 \in r_1.\text{reservoirs} \; (\; r_1.\text{geometry.within}(r_2.\text{geometry}) \;)$$

The trigger obtained when translating this constraint at the physical level uses the function sdo_inside provided by Oracle as follows.

```
create or replace trigger RiverReservoirsInside
before insert on River for each row
begin
  if exists ( select * from table(:new.reservoirs) r
    where sdo_inside(r,:new.geometry)='FALSE' )
  then
    raise_application_error(-20402,'Reservoirs must be located inside its river')
  end if;
end;
```

This trigger prevents the insertion of a river if the constraint is violated.

Space-Varying Attributes. A set of integrity constraints is generated when translating varying attributes such as depth in Lake (cf. Figure 2). Two of them verify that every value of attribute point is of spatial type point and is located inside the geometry of the lake. These triggers are similar to those given for attribute reservoirs above. Another constraint could be used to verify that the points are at least 1 meter from each other.

```
create or replace trigger LakeDepthDistance1m
before insert on Lake for each row
begin
  if exists ( select * from table(:new.depth) d1, table(:new.depth) d2
```

```
    where sdo_within_distance(d1.point,d2.point,'distance=1')='TRUE' )
  then
    raise_application_error(-20403,'Points must be at least 1 m. from each other')
  end if;
end;
```

Topological Relationships. Topological relationships constrain the geometry of the linked objects. For instance, in Figure 1 Overflow is a topological relationship of type intersect. This means that an instance of River may be linked to an instance of Flood only if their geometries intersect.

As can be seen in Figure 2, the topological constraint is lost in the transformation, only its underlying binary relationship is represented in the logical schema. A set of triggers is generated for ensuring the constraint. One of them fires upon insertions on Flood as follows:

```
create or replace trigger FloodOverflowTopological
after insert on Flood for each row
begin
  if exists ( select * from table(:new.riverRef) r,
  where not exists ( select * from table(r.column_value.floodRef) f
    where f.column_value.idFlood=:new.idFlood
    and sdo_overlaps(:new.geometry,r.column_value.geometry)='TRUE' ) ) )
  then
    raise_application_error(-20404,'Violation of Overflow topological relationship')
  end if;
end;
```

A symmetric trigger fires upon insertions of River verifying that the topological relationship is respected.

5 Conclusion and Future Work

Current practice in database design advocates the use of a conceptual model for capturing the data requirements of an application. The resulting conceptual schema is then translated into a logical and physical schemas targeted to specific implementation platforms (DBMSs or GISs). However, the main problem of this approach is that most of the semantics captured in the conceptual schema is lost in the translation process. The reason for this is the limited expression power provided by current data management software.

This paper presents a methodology that preserves the semantics of conceptual schemas using integrity constraints expressed at the logical and physical levels. Constraints at the logical level are expressed in first-order formulas that use the methods provided by the spatial and temporal data types. These constraints are then translated at the physical level either as check constraints or as triggers.

We gave examples of the methodology using MADS, a conceptual spatio-temporal model having powerful constructs for expressing the semantics of real-world applications. However, our methodology is generic and can be applied

to any conceptual model, e.g., Perceptory [1] or STER [9]. Obviously, as the expression power of the different conceptual models varies, it is necessary to first indentify the *implicit* integrity constaints built-in into the model, and then provide adequate translations at the logical and physical levels.

Another issue is to address *explicit* integrity constraints, specifying additional semantics that cannot be captured by the constructs of the conceptual model. These constraints are expressed using a specific language, e.g., OCL [11] for UML. The main problem is that it is not possible to realize an *automatic* translation of explicit constraints at the logical and physical levels, since these constraints may be arbitrarily complex. Therefore, a semi-automatic approach must be devised where the user is assisted in translating such contraints. This is an open research problem.

We are currently testing the performance of this approach in real applications. An important issue is to decide which constraints (among the thousands of medium-sized schemas) should be explicitly taken into account with our approach as it is practically impossible to include all of them in a schema. Further, although we validated our strategy in Oracle 10g using an object-relational model, we plan to support other platforms such as SQL Server and ArcView.

References

1. Y. Bédard. Visual modeling of spatial databases: Towards Spatial PVL and UML. *Geomatica*, 53(2):169-185, 1999.
2. S. Cockcroft. A taxonomy of spatial data integrity constraints. *GeoInformatica*, 1(4):327-343, 1983.
3. J.H. Doorn, L. Rivero, editors. Database Integrity: Challenges & Solutions. Idea Group Publishing, 2002.
4. M.J. Egenhofer, E. Clementini, P. Di Felice. Topological relations between regions with holes. *Int. Journal of Geographic Information Systems,* 8(2):129-142, 1994.
5. M. Minout, C. Parent, E. Zimányi. A tool for transforming conceptual schemas of spatio-temporal databases with multiple representations. In *Proc. of the IASTED Int. Conf. on Database Applications, DBA'2004*, 2004.
6. C. Parent, S. Spaccapietra, E. Zimányi. Spatio-temporal conceptual models: Data structures + space + time. In *Proc. of the 7th ACM Symposium on Advances in Geographic Information Systems, GIS'99*, pages 26–33, 1999.
7. C. Parent, S. Spaccapietra, E. Zimányi. The MurMur project: Modeling and querying multi-representation spatio-temporal databases. To appear in *Information Systems*, 2005.
8. C. Parent, S. Spaccapietra, E. Zimányi. *Conceptual Modeling for Traditional and Spatio-Temporal Applications: The MADS Approach*. Springer, 2005, to appear.
9. N. Tryfona, R. Price, C.S. Price. Spatiotemporal Conceptual Modeling. In M. Koubarakis *et al.*, eds., *Spatio-Temporal Databases: The Chorochronos Approach*, Chapter 3, pp. 79-116. LNCS 2520, Springer, 2003.
10. C. Türker, M Gertz. Semantic integrity support in SQL:1999 and commercial (object)-relational database management systems. *Very Large Databases Journal*, 10(1):241-269, 2001.
11. J. Warmer, A. Kleppe. *The Object Constraint Language Second Edition: Getting your Models Ready for MDA*. Addison-Wesley, 2003.

Using Image Schemata to Represent Meaningful Spatial Configurations

Urs-Jakob Rüetschi and Sabine Timpf

Department of Geography, University of Zürich
{uruetsch, timpf}@geo.unizh.ch

Abstract. Spatial configurations have a meaning to humans. For example, if I am standing on a square in front of a building, and this building has a door, then this means to me that this door leads into the building. This type of meaning can be nicely captured by image schemata, patterns in our mind that help us making sense of what we perceive. Spatial configurations can be structured taxonomically and mereologically by means of image schemata in a way that is believed to be close to human cognition. This paper focuses on a specific application domain, train stations, but also tries to generalise to other levels of scale and other types of spaces, showing benefits and limits.

1 Introduction

If semantics is the study of meaning [6], then spatial semantics is about the meaning of space and spatial configurations. Our research group investigates human wayfinding in public transport. To this end, we need a semantic model of train stations, a model that represents what the spatial configuration of a station *means* to the traveller. Unfortunately, traditional GIS data models are so different from human cognition that they hardly capture the meaning of the geometries they represent. Yet there is ample evidence in the literature that the precise geometry is far less important to human cognition than the coarse spatial layout, that is, how spaces are nested inside one another and connected to one another; see, e.g., [13,11,16].

This paper examines spatial image schemata [7] for the meaningful representation of train stations, our application domain, and architectural spaces in general. We build on the hypothesis that image schemata can capture the semantics of complete spatial configurations, not merely of individual situations (like "the pawn is on the chess board"). Our approach is based on an idea presented in [22]. It differs from previous image-schemata-related work in the GIS community (see, e.g., [14,10,19,20]) in that it does not only look at individual schemata, but at how they interact to build complex spatial structures. It is also different from work by Egenhofer and colleagues about topological spatial relations (e.g., [2]), which is also qualitative but does not follow the synthetic and object centered approach we do. Nevertheless, important qualitative spatial relations will emerge from the image schema based approach, as we will see.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1047–1055, 2005.
© Springer-Verlag Berlin Heidelberg 2005

This text is organised as follows. Section 2 introduces spatial image schemata and how they help with semantics in the context of our application domain. Section 3 shows how instances of these schemata relate to each other. Section 4 identifies dependencies among image schemata and summarises these in the form of consistency rules. Section 5 tries to generalise from our application domain to other levels of scale, indicating potentials and limits. Finally, section 6 can confirm our hypothesis in a discussion of the use of image schemata as presented in this text; it also adds a quick look at aspects of implementation and integration into existing systems.

2 Spatial Image Schemata

When investigating train stations, we observe that there are some spatial elements that occur repeatedly. For example, most stations have a station hall, containing such things as timetables, waiting areas, doors connecting it to the station square, and some platforms. These elements can be abstracted as instances of image schemata.

Image schemata originated in philosophy and linguistics, where they refer to recurrent patterns that help us making sense of our perceptions and actions. They consist of parts and relations that may or may not match perceptual input [7,12]. Many of them are inherently spatial [3]. In this paper, we consider the schemata in table 1. A station hall, for example, can be seen as an instance of a CONTAINER: it has a boundary, namely some walls, which separate the inside of the hall from the outside, thereby inducing a relation of containment.

Table 1. Spatial image schemata with parts and relations

Image Schema	parts	induced relation
COLLECTION:	none	part-of (element-of)
SURFACE:	the surface	part-of (on/off)
CONTAINER:	inside, outside, boundary	part-of (contains)
LINK:	none	connect
GATEWAY:	the (general) door	connect
PATH:	source, destination, trajectory	connect, (left/right, not considered here)
OBJECT:	the object	(not considered here)

Image schemata are useful for spatial semantics:

1. they have instances that are located in space and structure space,
2. they afford [5] some activity, that is, they communicate some meaning, and
3. they combine in various ways to describe complex spatial configurations.

This is very different from spatial models that build on points, lines, and polygons that are also located in space but have certainly no immediate meaning.

3 Spatial Configurations

Image schemata do not occur in isolation, rather their instances are related to each other in a way that adds structure and meaning to the space they constitute. For example, a building instantiates the CONTAINER schema and thus can contain other instances, such as objects, doors connecting the inside and the outside (GATEWAY), and rooms (CONTAINER in CONTAINER).

We investigate the two generic relations indicated in table 1: CONTAINER, SURFACE, and COLLECTION all induce a *part-of* relationship, whereas GATEWAY, PATH, and LINK induce a *connection* relationship. This gives rise to the following classification of spatial image schemata:

1. collecting schemata (→ partonomy)
2. linking schemata (→ network structure)
3. other schemata (in this paper only the OBJECT schema).

Instances of collecting schemata result in a partonomy, which can be formally represented as a poset (partially ordered set; Fig. 1). This poset is, in general, not a tree, for multiple containment is possible (see also [1]). Whether or not it is strict (i.e., not reflexive) is largely a matter of taste. We will consider it strict, that is, a collection is not part of itself. More interesting is the question, if the poset is a lattice. This question is investigated in Sect. 4.

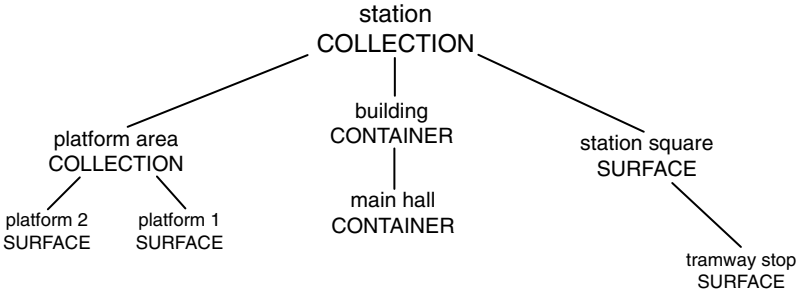


Fig. 1. A simple partonomy of instances of the schemata indicated in capitals

The linking schemata create connections between instances of the collecting schemata, that is, between existing elements in the partonomy (or poset). The link itself can be considered yet another element in the partonomy, with two "parents," namely the two other elements it links together. Adding linking elements to the partonomy in this way results in an enlarged partonomy, but the structure is still that of a poset. If we look at the connections created by these linking elements, we find that there are networks within the partonomy. This is highlighted in Fig. 2. We may think of the partonomy as a "vertical" graph, whereas the parts of it that have a network structure can be thought of as "horizontal" graphs.

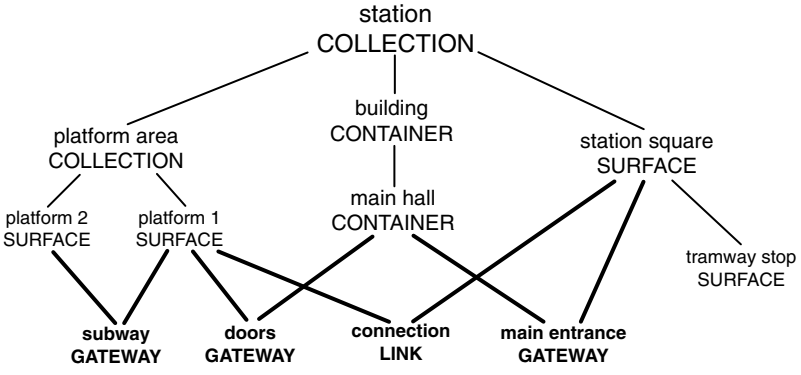


Fig. 2. The simple partonomy from Fig. 1, extended with network connections, i.e., instances of linking schemata

Connections can be directed (like an escalator) or bi-directional (like a door). Since only the PATH schema implies a direction, it is recommended to store a direction (or the fact that a link is bi-directional) explicitly in an attribute to the schema instance. Such attributes can be used to represent arbitrary auxiliary information that helps to distinguish one schema instance from another.

The PATH schema instances are special in another way besides directedness: they have a trajectory as one of their vital components. For example, a typical PATH in our application domain may lead from the entrance, across the hall, down the escalator, and into the shopping area. It can be represented as an alternating sequence of collecting schema instances and linking schema instances. But there are other types of PATH instances, where the trajectory has an existence on its own, like a street in a town. Here we stick with PATH instances of the former kind, the latter is quickly looked at in Sect. 5.

4 Consistency Rules

Instances of the image schemata cannot be at arbitrary positions in the poset and relative to each other. Rather, some consistency rules have to be followed for the structure to be meaningful:

1. linking elements and OBJECT instances must be minimal in the poset (they can't contain)
2. linking elements can't be maximal (they must be part of what they link)
3. each CONTAINER must have at least one GATEWAY or LINK (otherwise it is not recognisable as a CONTAINER)

These rules are a direct consequence of the image schemata; therefore, any violation results in a structure that is no longer meaningful, or at least confusing, because normal human reasoning is disturbed. Besides these three rules, we may also require:

4. there is a greatest element, the study area
5. the poset of schema instances must form an upper semilattice

Rule 4 is useful, for it states that there must be an element that corresponds to the entire study area, but it is not implied by the image schemata. Rule 5 (which implies rule 4) would be useful, because a lattice is a more specific structure than an arbitrary poset. But the simple example of two rooms (CONTAINER) that are connected by two doors (GATEWAY) is a realistic scenario and a counter-example (Fig. 3). However, any poset can be turned into a lattice using a procedure known as normal completion [8]. This procedure adds some "artificial" elements such that the lattice conditions hold. Other than in [9], these new elements are not necessarily the intersection of existing elements. Rather, they represent a COLLECTION that is not explicitly modelled. In a sense, it represents a plural, such as "room A and room B are connected by *some doors,*" and by going down in the lattice, these "doors" are explicitly mentioned (see Fig. 3).

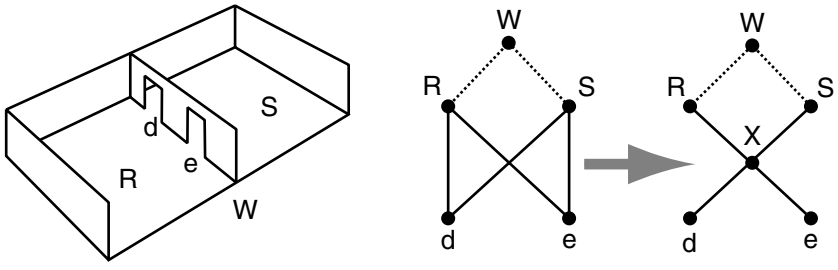


Fig. 3. If in an environment W two rooms R and S are connected with each other by two doors d and e, then the resulting poset is not a lattice. However, it can be turned into a lattice by adding an "artificial" element, X. The interpretation of this new element is "some ⟨schema instances⟩," in this case "two doors".

5 Change of Scale

So far, we looked at a train station, which is at an environmental scale level [17]. Our image-schematic approach allowed us to decompose this Zubin type D object [15] into type C objects, the platforms, halls, squares, etc. Now it is interesting to see if this approach also works at other scales.

At a smaller scale. are the objects within the station. Examples include: signs, clocks, letter boxes, information panels, posters, but also relations like x is attached to y. They exist at figural and vista scales [17], and they are type A objects according to Zubin. The following schemata are instanciated (with an example in parentheses): CONTAINER (letter box), GATEWAY (its slot), SURFACE (poster), OBJECT (clock), LINK (poster attached to poster panel), COLLECTION (group of poster panels). That is, we have the same schemata (except for PATH), but their interpretation differs somewhat:

1. CONTAINER is no longer enterable (for humans)
2. SURFACE is typically vertical and not "stand-on-able"
3. LINK means attachment by construction

Some pioneering work and practical experience can be found in the development tools for *text adventure games*. These engines provide an elaborate machinery for dealing with objects in space from a semantic rather than geometric point of view, for only minimal geometric information is required for producing textual descriptions and parsing textual input. It is particularly interesting to note that the CONTAINER and SURFACE schema are explicitly present at the figural and vista scale level. Unfortunately, these engines are far less adept at environmental scale levels.

Here is an example from a game written using the Inform system [18]:

```
Object -> wicker_cage "wicker cage"
  with  description "Its a small wicker cage"
  has   container open openable transparent;
```

Another system is TADS,[1] for which the following example is written:

```
+ cupboard: ComplexContainer, Heavy 'old cupboard'
  "The cupboard is a battered old wooden thing..."
  subContainer : ComplexComponent, OpenableContainer { }
  subSurface : ComplexComponent, Surface { }
;
```

Keywords such as "container" or classes such as "Surface" in the examples above are used by game authors to tell the game engine about essential properties of the objects that are part of the game.

At a larger scale. we find the station in its (urban) environment. The structure of this typically outdoor space is dominated by the network of streets and squares, as opposed to the rooms and gateways we had before. The schema that is most prominent in this new type of space is the PATH in its second interpretation (Sect. 3): here the PATH schema's trajectory has an identity on its own, it is manifest in the form of a pavement, of street markings, and so on. These PATH instances connect places, mostly squares, which are instances of the SURFACE schema. The GATEWAY instances serve as links to the lower scale level; they correspond to such things as entrances into buildings.

The match between the schemata from table 1 and the new situation is not as perfect as it was at the lower scale level. But this is not as much due to the change in the scale level, but rather to a major shift in the qualitative structure of this larger space: away from nested spaces towards a classical network of streets and places. In [23], these two types of spaces are referred to as *scene space* and *network space*, respectively.

[1] http://www.tads.org/

Knowing the scale level is important when working with image schemata, for it determines the interpretation of the schemata. CONTAINER instances in figural space are certainly not enterable for a human. PATH instances are "walk-along-able" both in scene space and in network space, but the meaning of the trajectory component changes. Consequently, we have to record the scale level as an attribute to the schema instances. To this end, we can use one of the many classifications of space, such as the one by Montello [17] (see [3] for an overview). In any case, the scale attribute has to respect another consistency rule:

6. x part-of y implies $\text{scale}(x) \leq \text{scale}(y)$,
   x and y arbitrary elements in the partonomy.

6 Discussion and Conclusions

Starting from an observation of train stations, we arrived at a small number of spatial image schemata that can be used to structure space. They are useful to represent the semantics of spatial configurations, because they relate directly to human cognition. The novelty of the approach lies in the recognition that the relations induced by the image schemata can be exploited to build formal structures (namely posets) and thus making meaningful space accessible to computational procedures. Three essential semantic relations are explicitly present: *type-of* by the assignment of instances to one of the image schemata, *part-of* by the partonomy, and *connect* as instances of the linking schemata. The schema assignment helps with specifying the general part-of relation: for CONTAINER it is "in" or "inside," for SURFACE it is "on," and for COLLECTION it is just the general "part-of."

Another semantic relation is synonymy. It would be interesting to extend the concept of synonymy to spatial configurations (so we probably then better refer to it as isomorphy): when are two spatial configurations to be considered synonymous? A simple approach is to require that two configurations are isomorphous as posets and that the corresponding elements instantiate the same schemata. But that is an area for future research, including empirical testing.

Unfortunately, there is no established data structure to represent general posets. From a mathematical point of view the most elegant solution is to work with linear extensions of the poset. This approach is fine for posets of small dimension [21], but determining the dimension of a poset is NP-hard [24]. Given that the posets arising from our application domain are rather small, this is no serious limitation. Nevertheless, we should try to exploit the properties of the phenomenon to be represented as a poset. For example, if we assume that every element in the poset, even linking elements, has some spatial extent, then pairwise intersections result in other, smaller extents. Each original element in the poset is then a set of one or many of these small spatial extents and order testing is simply testing for set inclusion. Moreover, this defines an embedding of the poset in an n-dimensional cube, that is, in a particular class of lattices.

The proposed model can be a useful supplement to existing GIS by using it as an additional semantical layer associated with a spatial data layer. This

requires a partial mapping between polygons in the spatial layer with elements in the semantical layer (the partonomy) such that the software can easily switch from geometry representation to cognitive semantics and vice versa.

In summary, the proposed method based on image schemata captures important semantical aspects of spatial configurations and nicely integrates these over a range of scales, though some care has to be taken when a model comprises different scale levels. This confirms the hypothesis set forth in the first section and shows that image schemata are a useful tool when working with the meaning of spatial configurations.

Acknowledgments

The authors acknowledge Jürg Nievergelt for suggesting the cube embedding approach for representing posets and the anonymous reviewers for their helpful comments. Financial support was provided by the Swiss National Science Foundation.

References

1. Alexander, Ch.: A city is not a tree. *Architectural Forum* **122**(1 and 2), 1965.
2. Egenhofer, M. J. and R. D. Franzosa: Point-set topological spatial relations. *International Journal of Geographical Information Systems*, **5**(2), 161–174, 1991.
3. Freundschuh, S. M. and M. Sharma: Spatial Image Schemata, Locative Terms, and Geographic Spaces in Children's Narrative: Fostering Spatial Skills in Children. *Cartographica* **32**(2), 38–49, 1996.
4. Freundschuh, S. M. and M. J. Egenhofer: Human Conceptions of Spaces: Implications for Geographic Information Systems. *Transactions in GIS* **2**(4), 361–375, 1997.
5. Gibson, J. J.: *The Ecological Approach to Visual Perception*, Lawrence Erlbaum Associates, London, 1986 (first published in 1979).
6. Heller, J.: Semantic Structures. In *Knowledge structures*, edited by D. Albert. Springer-Verlag, New York, 1994.
7. Johnson, M.: *The Body in the Mind. The Bodily Basis of Meaning, Imagination, and Reason.* The University of Chicago Press, 1987.
8. Kainz, W.: Application of Lattice Theory to Geography. In *Proceedings of the 3rd International Symposium on Spatial Data Handling*, Sydney, 1988.
9. Kainz, W., Egenhofer, M., and Greasly, I.: Modelling Spatial Relations and Operations with Partially Ordered Sets. In *International Journal of Geographical Information Systems*, vol. 7, pp. 214–229, 1993.
10. Kuhn, W. and A. U. Frank: A Formalization of Metaphors and Image-Schemas in User Interfaces. In *Cognitive and Linguistic Aspects of Geographic Space*, Kluwer Academic Publishers, 1991.
11. Kuipers, B. J.: The "map in the head" metaphor. *Environment and Behavior* **14**(2), 202–220, 1982.
12. Lakoff, G.: *Women, Fire, and Dangerous Things. What Categories Reveal about the Mind.* The University of Chicago Press, 1987.
13. Lynch, K.: *The Image of the City*. The M.I.T. Press, Cambridge MA, 1960.

14. Mark, D. M.: Cognitive Image-Schemata for Geographic Information: Relations to User Views and GIS Interfaces. *Proc. GIS/LIS'89, Orlando*, 1989.
15. Mark, D. M. et al.: Languages of Spatial Relations: Initiative 2 Specialist Meeting Report, NCGIA Technical Paper 89-2, May 1989.
16. Montello, D. R.: The geometry of environmental knowledge. In *Theories and methods of spatio-temporal reasoning in geographic space*, edited by A. U. Frank and U. Formentini, Springer-Verlag, Berlin, LNCS 639, 1992.
17. Montello, D. R.: Scale and multiple psychologies of space. In *Spatial Information Theory. Proceedings COSIT'93*, Springer LNCS 716, 1993.
18. Nelson, G.: *The Inform Designer's Manual*, 4th edition. The Interactive Fiction Library, St. Charles, Illinois, 2001.
19. Rodriguez, M. A. and M. J. Egenhofer: Image-Schemata-Based Spatial Inferences: The Container-Surface Algebra. In *Spatial Information Theory. Proceedings COSIT'97*, Springer LNCS 1329, 1997.
20. Raubal, M.: *Wayfinding in Built Environments. The Case of Airports*. IfGI prints, vol. 14, University of Münster, Germany, 2002.
21. Schröder, B.S.W.: *Ordered Sets: An Introduction*. Birkhäuser, Boston, 2003.
22. Rüetschi, U. J. and S. Timpf: Schematic Geometry of Public Transport Spaces for Wayfinding. In *Geoinformatik und Mobilität. Proceedings Münsteraner GI-Tage*, Münster, Germany, 2004.
23. Rüetschi, U. J. and S. Timpf: Modelling Wayfinding in Public Transport: Network Space and Scene Space. Proc. *Spatial Cognition IV*, Lake Chiemsee, Germany, 2004.
24. Yannakakis, M.: The Complexity of the Partial Order Dimension Problem. In *SIAM Journal on Algebraic and Discrete Methods*, **3**(3), 1982.

Collaborative geoVisualization: Object-Field Representations with Semantic and Uncertainty Information

Vlasios Voudouris, Jo Wood, and Peter F Fisher

giCentre, Department of Information Science, City University, London, UK
{vv, jwo, pff1}@city.ac.uk

Abstract. Techniques and issues for the characterisation of an object-field representation that includes notions of semantics and uncertainty are detailed. The purpose of this model is to allow users to capture objects in field with internally variable levels of uncertainty, to visualize users' conceptualizations of those geographic domains, and to share their understanding with others using embedded semantics. Concepts from collaborative environments inform the development of this semantic-driven model as well as the importance of presenting all collaborators' analysis in a way that enables them to fully communicate their views and understandings about the object and the field within it. First, a conceptual background is provided which briefly addresses collaborative environments and the concepts behind an object-field representation. Second, implementation of that model within a database is discussed. Finally, a LandCover example is presented as a way of illustrating the applicability of the semantic model.

1 Introduction

Methods for geographic representation are frequently classified into raster and vector models. Conceptualised as fields and objects, respectively, they are frequently considered as dual approaches for modelling geographical phenomena [2]. Object approaches define space bounded and filled by using Euclidean Boolean objects primarily with polygonal boundaries, enabling the description of specific entities. In contrast, fields do not populate the space with entities, but rather characterize space by locational properties that are usually regarded as continuous phenomena. Thus in field perspective a value selected from the attribute domain is mapped. These two approaches have been widely viewed as complementary for describing the world, but non-overlapping.

It is the contention of the research presented here that it is possible to link the two models bridging between objects and fields, and that there are conceptual and analytical advantages in doing so. Any hybrid model must integrate the two data structures at the conceptual/object level. In this paper we propose a model which enables the identification of objects from a continuous phenomena with the retention of variable levels of uncertainty within the object and with semantic description of the user's understanding of the object. The implementation is grounded in the raster data structure. Thus the model attempts to unify the conceptualization methods rather than

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1056 – 1065, 2005.
© Springer-Verlag Berlin Heidelberg 2005

the representation. In other words, the main purpose of this paper is to describe the means of linking fields (spatially represented as raster) and object conceptualizations of geographical space as a means for conveying understanding in collaborative work. The approach is based on mapping fields to entities represented as raster objects. Attached to the objects are semantic and uncertainty information that depict observer's conceptualization of the field. Thus, we describe how the object-field approach which combines these two models can be adapted to represent the semantic uncertainty inherent in collaborative GI modeling.

The conceptual background associated with field and object models as well as understanding of collaborative environments is outlined in more detail in the following section, and the third section discusses the data structure required for the proposing model. One application of the approach is presented, and the paper concludes with a discussion of areas for further research.

2 Object Field Representations Within Collaborative Environments

2.1 Conceptual Background

The entry of geoinformation into mainstream desktop and mobile computing along with the increased use of the web creates the possibility to bringing a range of people, a range of data and a range of resources to bear on a problem. However, bringing a range of people together requires the managing of different types of interaction, namely between systems and humans and among humans-systems-humans, in order to effectively support co-operative work by individuals who bring different perspectives to (spatial or aspatial) tasks. The collaborative working is only possible by developing or extending visualization methods and tools that support the idea of sharing knowledge, constructing knowledge and exploring data collaboratively. (Geo)visualization can provide the techniques for developing visual methods for enabling the building of concepts/interpretations form observational data and connecting these concepts to each other and for exploring complex data sets by relating different data for additional information [4]. Furthermore, given that the most effective work on problems is frequently conducted by a group of people, the need to connect people together is essential. However, connecting people is not an easy task because not only do technical issues need to be addressed but there are also semantic issues. For example, different people may respond with different interpretations of the same visualised information. These interpretations are influenced by both the social and technical environment within which the collaborative work is conducted. Thus, one of the key issues in collaborative work is to employ and argue about ideas and views. With the GI field it is likely that the idea and views are presented as text, graphics (including statistical and map based representations), images (photographic) and tabular data. Given the inherent and time complexities such as social interactions, differing semantics and a variety of different and possibly conflicting working practices among collaborators it is particularly important to provide a mechanism that enables collaborators to share their analysis as well as to argue about the analysis at

different scales using a variety of evidence to support their arguments which can be reported as spatial multimedia.

The object space of the model, for example, can be used as a means of conceptualization reporting to manage different user-interpretations in collaborative GI work. Thus, collaborators can transfer not only raw data but also understanding and interpretations. Semantics are important in driving the way the analysis/map should be understood/read. This is of particular importance when a group of people try to identify and detect the dynamic behaviour of fuzzy objects caused by time of observation and/or particular observer. It is arguably difficult to detect the dynamic behaviour of fuzzy objects requiring the use of prior knowledge and semantics. In collaborative work, observers should be able to fully communicate the associated semantics and knowledge used for their dynamic analysis of the fuzzy objects. Thus a mechanism capable of encapsulating both *crisp concepts* represented, for example, as semantic objects and *fuzzy spatial extents* represented as fields is required. This is of particular importance because uncertainty is not only associated with positional and coordinate accuracy, but also with object definition and thematic vagueness – intention and extension [8].

By developing semantic-based models it is anticipated that different analysis among collaborators will be communicated ideally through the provision of some sort of agreement over spatial objects and their meaning as well as by adopting similar ontology by using the concept of equivalency table. In other words, the model should provide a way of sharing conceptualizations and semantics that drive how others formulate and understand these conceptualizations within a certain collaborative application context as 'more often than not the users are not very specific about their definition of their objects and object classes, etc. ' [8]. Such ill-defined definitions can hinder appropriate collaborative analysis and conclusions.

[10] defines conceptualization as "a system of concepts or categories that divide up the pertinent domain into objects, qualities, relations and so forth". A specification of a conceptualization or ontology specifies the objects that are to be investigated and the methodology to be adopted [6]. Thus, by just representing the world in a paper or digital environment we necessarily lose appropriate information that enables users of the information to fully understand the producer's knowledge. This results partly in representing or interpreting the same phenomenon differently leading to problematic sharing of knowledge especially in collaborative environments with multi-disciplinary participants. This is because each collaborator brings his/her own disciplinary view and work practises to the geographic information that is used. Data models that permit the updating and assessment of semantic information that do not necessarily describe only measurable aspects such as map unit, positional accuracy at different level is therefore a key need. The approach we argue for here requires a data model that enables knowledgeable users to improve data quality by interactively creating and attaching semantics and spatially variable uncertainty to objects. In other words, collaborators will interactively create semantic entities out of observational data based on their perceptions and concepts.

However, for the identification of the features and the associated semantics, it is important not to specify strict rules in order to make the model applicable to a variety of application domains. Each domain could then apply appropriate rules for describing the semantics that are relevant. The focus is to define a way of encoding

spatial boundaries of the objects, the representation of the objects and to establish the relationships among the objects and between objects and fields. In is also important to note that for the integration of semantics, the concept of *Look Up Tables* is adopted [1]. This is a way of addressing the differences in semantics by different collaborators using mutually agreeable semantics and by adopting multimedia/interactive techniques for sharing semantics, the collaborators can interactively build an equivalency table that is tailored to their needs and application requirements. For example, when considering LandCover, collaborators can interactively build a digital report by attaching/georeferencing semantic and uncertainty objects at mapset, object class and spatial object level as the collaborative discussion progresses. Thus, the digital report which is constructed interactively by the collaborators is made up of *georeferenced semantic and uncertainty objects attached/associated with raster objects* (see figure 1).

For the definition of the spatial-conceptual objects, the approach proposed [3] will be adopted as a starting point. Fields and objects are mapped together by 'defining a continuous field in which locations in a field -space are mapped to spatial objects in an object-space'([3], p. 4). That is the identification of point, line and area objects represented by a set of pixels in the raster model. The data structures appropriate for storing such models will depend in part, on the types of relationship that exist between the field and object representations. For example, a simple case might be the 1-to-1 relationship where each tesseral (raster) cell is uniquely associated with a single object representation. More complex relationships can be modelled if several cells can share the same object representation (many-to-1), or even a collection of objects (many-to-many). See [3], [12] and [13] for more details.

It is also important to establish not only the cardinality of the relationships but also data structure mechanism or representing the relationship. According to [8], two ways exist for linking objects to cells:

- Each cell has a ID/label indicating to which object/(meta)object it belongs. By using 'normal' raster algorithms, each cell is questioned and then the geometry of the object is found
- Or, each object holds a reference to the cell that represents the object by using [linked] list [11].

In other words, the objects are found through their location (first case) or the geometry is found through the objects (second case). In both cases once the objects are found, the metadata attached to them can be explored. Thus, in adopting the type of relationships it is important to analyse the way people seek to find objects. As the underlying assumption for this paper is that the semantic objects act as means of transferring conceptualizations out of field representation by identification of raster objects the first approach will be adopted here. Furthermore, in a collaborative environment that uses spatial user interfaces it is important to allow the observers to specify/identify the spatial extent of field-based concepts they identify/conceptualise and then to attach attribute data as part of the object representation. Additionally, this feature identification is important for searching, extraction and feature-level management by minimizing the foreign keys required [9].

2.2 Formalizing the Approach

It is important to point out that object-field model does not require a new data structure for raster representations, but it is largely based on a different interpretation of the raster model that actually *extends* the conventional raster model. The extended model has the advantage of offering richer and more realistic semantics at the cost of more calculations and augmented storage capacity in comparison to the non-extended conventional raster model.

Figure 1 provides an overview of the implementation of the semantic model in the Unified Modeling Language (UML), an object-oriented visual modeling language. Each rectangle represents a class, a java class for example which is a database representation of the concepts counterparts aforementioned. Methods and fields associated with the class are not shown in the diagram for simplicity. Lines with empty diamond represent aggregation relationship\weak containment; lines with filled diamond represent composition\strong containment, lines with triangle represent generalization/inheritance relationship and simple lines represent associations. The arc represents that each cell is associated with a list of cell id's in its 'object'. And each cell has two roles, namely the field and the object.

The approach presented enables the technological aspect to inform the ontological one by representing observation data and attaching associated metadata and uncertainty in one integrated semantic data model (see figure 1) thus specifying map cover features (as a set of cells) that result in certain conceptualizations where the precise nature is described in metadata (semantic and uncertainty objects).

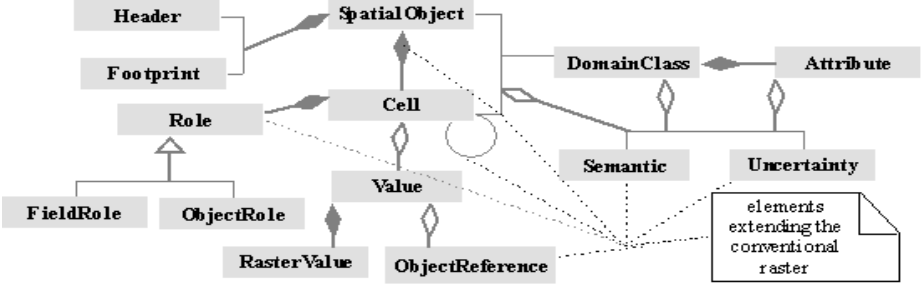


Fig. 1. Object-Field representation with semantics and uncertainty at different level

Based on the above diagram, it is clear that three levels of uncertainty and semantic information exist. One level is used to describe object classes (DomainClass) and the second lower level is used to describe spatial objects (SpatialObject) and the third the map unit-smallest mappable area, raster cell in the diagram presented. Furthermore, a raster cell can have a number of values with different semantics and uncertainties attached while the set of locations is only associated with one domain object. The value objects permit Boolean or degree of membership, which can be used to represent fuzzy conceptualizations. However, Boolean members can be used to generate binary representations that may be necessary towards the end of a collaborative discussion

when decisions need to be taken and fuzzy conceptualizations are not permitted. This is particularly important to time-critical applications.

Through this integrated approach it will be possible to identify where uncertainty informs the spatial description of the object and how the semantics are informed by this. Especially the raster-level semantics and uncertainty can form objects that justify their existent on exceptional circumstances by overriding class-level uncertainties and semantics. For example, the identification of peaks not justified by mathematical calculations rather by expert knowledge of the study area. In this case, semantics can be used as additional classification conditions. By recording these information Dempster-Shafer theory of evidence offer a mechanism to combine multiple pieces of evidence that include notions of semantics and uncertainty. Class-level semantics are important to systematically define criteria for mapping classes in order to avoid making pixel-level metadata worthless (see [1] for examples of problematic class definitions of land cover cases). Furthermore, this pixel-level metadata can be used as data at the spatial-object level. For example, let's assume that we are trying to identify the ethnicity of a group of people by using just their full names. However, the same first or/and last name can be used by many different countries (for example, Greece and Cyprus). But if we combine the possibilities/uncertainties [of being Greek or Cypriot] attached to the first and last name (which act as the lowest mapping unit[pixel] for this example) we can generate a combined possibility for the full name (which act as spatialObject, as it represents an individual in our context]. These combined possibilities/uncertainties give us additional data for categorizing/ classifying (DomainClass) every single person in our study. To move the demonstration further, we can attach semantic data describing the letters used for the last and the first names. This semantic data is also additional data at the full-name level as it represents some cultural or historic data at the country level. By making metadata useful for the end user, we increase the possibility of attaching metadata because users are getting back directly the benefit of the metadata at some stage of their work.

The model supports analysing and querying of the objects, their states including their spatial extent and specific thematic values, semantic and levels of uncertainty. Observers can attach three levels of metadata - levels of uncertainty and semantics by domain conceptualized objects to raster objects (SpatialObject) interactively. In other words, observers will attach semantic and uncertainty objects by selecting or specifying a region represented as a set of cells. In turn, by interactively querying the objects it will be possible to extract and analyse only those objects or part of those objects that fell within a given certaint level (for example, in retail application, one can ask for catchments areas that have a certainty level of 0.7 or above). As a result, non-relevant areas are excluded from the computation.

3 Application Example: LandCover Mapping in the UK

A central quality of the framework aforementioned is the semantic and uncertainty information attached to 'simple' geographic elements-raster cell in the case- as well as to the definition of the classes to which the individual objects belong.

We will illustrate the applicably of the model using LandCover data. The application is constructed around a number of basic concepts of semantic and uncertainty multimedia (uncertainty represented as text, tables, graphics or audio) and visualization objects:

- Linking of multiple objects in a constant geographic framework aids in the analysis and interpretation of different conceptualizations, including semantics.
- Any number of spatial linking multimedia components can be used and related as a evidence supporting an interpetation.
- Fields and objects can be interactive explored and updated according to exploratory information needs of the collaborative observer. The interactivity is bidirectional meaning that the users have a number of alterative entry points.
- Geographic information is recorded at the field level and multiple attributes are associated with individual map units or regions representing and categorised as semantic and uncertainty interactively by the user meaning that *the human supervisory* role is strictly maintained.

The example presented below will demonstrate that by integrating the representation of geographic phenomena, semantic and uncertainty (in conceptualization) is possible to communicate the producer's conceptualization/knowledge and to introduce invariable and fixed definitions of fuzzy units (LandCover features can be regarded as fuzzy objects) within collaborative LandCover cases. A second effect can be that fact that metadata can also inform the information systems that process the data in order to produce the same land cover map or demonstrate the way the land cover map is/was produced. Both of them are key factors in designing semantic-driven collaborative systems that effectively enable the sharing of ideas and geo-methods. Jung *et al* [7] note that 'formal descriptions of the semantics of an operation to achieve a performance as similar as possible when applied to different data types/geometric representations' (p.3) is required in order to address not only overlay errors but also to develop a set of hybrid operations that are data structure/geometric representation independent. The need to communicate analytical results effectively requires the employment of abstract operations on geodata and metadata. By transferring metadata explaining the conceptualization process of the collaborator producing the analysis, observers may better cope with the problem of cognitive overload in understanding the difference between datasets.

The example is largely based on the work undertaken by Comber *et al* [1]. According to them remapping of land use and landcover is driven by changes in the use or cover itself. However, the resulting remapping is also influenced by the methodology employed and by the political initiative. The resulting ontological inconsistencies present difficulties for the monitoring of environmental change between two land cover mappings, for example (i.e. LCM1990 and LCM2000). Tracing these kinds of inconsistencies, due to the revised methodology or due to the changes in the phenomena being measured, in a collaborative environment add additional sharing/communication difficulties. Semantic-driven data models can effectively support the communication of geoanalysis by taking into account ontological inconsistencies and by presenting and integrating the previous and the new approaches as 'semantic objects'.

In the land-cover mapping, inconsistency (in a particular context) can be defined as 'whether the information for a particular LandCover object (in this case LCM2000 parcel) is inconsistent with the cover types within a parcel in LCM1990 when viewed through the lens of the Spectral and Semantic LUTs – Look Up Tables' ([1], p.4). This inconsistency is caused either by error or by an actual change on the ground. By recording pixel-level semantic and uncertainty information it is possible to analyse the distribution of different types of inconsistency and to make statistical predictions about the parcels. Thus, the pixel-level uncertainty and semantic information can inform or define a region [set of pixels]. By recording a number of 'history semantic objects' it would be possible to question, analyse and associate the changes in the semantic and uncertainty objects which will enable us to identify semantic inconsistencies. This means that temporal analysis of objects could be carried out more effectively. For example, the pixel-level information can inform the object/region creation/transformation: Not only the spatial extent of the region but also the cumulative uncertainty and semantic information that is provided by the pixels. Thus data at one level can act as metadata for another layer (see section 2.2), and per-pixel data can be reviewed to give a percentage of the set of locations that can be designated as a woodland category, for example. Alternatively fuzzy memberships of each and every pixel(data at one level) can be used as metadata to the usual hard classification- percent correctly classified accuracy metadata at the layer level.

Furthermore, there may be cases which produce, for example, the non-existence of any woodland in and area where woodland is known to exist. However, we do know that the National Forest exists and so based on this information we can override the result produced by the [supervised] algorithm. Overriding the results without attaching semantic information could lead to poor communication of information because the producer does not communicate effectively all of his/her knowledge. This leads to errors in subsequent interpretation and analysis by others as the rational has been lost. For example how classes are interpreted from the original imagery, how this may vary with region and what reliability may be placed on different classes. The users therefore are missing information which is possibly the most valuable [5]. The lack of local and detailed information can be captured by the semantic and uncertainty objects thus 'the map data and the alternative interpretations of the map unit are at their most useful when both can be used together within a GIS' [5]. The quality of the model proposed here is that all the data (spatial and aspatial, including graphic and tabular data) are fully integrated enabling LandCover reports and maps to be built together. Furthermore, when a group of users analyse two different datasets of land cover to determine environmental change, for example, the model can be used to gradually improve and link the two datasets resulting in a data, information and interpretation rich model.

By recording pixel-level and regional-level semantic information, space complexity is increased which leads to an increase in time complexity. As has been noted previously, the object-field representation combines the two perspectives in a way that permits a number of competitive spatial alternatives to be explored by presenting a wealth of results/information that can be interactively explored [3]. Thus, the space complexity can be regarded both as its disadvantage and as its advantage. It is also hoped that the object/feature-based processing approach will minimize the time complexity.

4 Conclusion and Next Steps

We believe that as a fully shared ontology (same scale, same purpose at the same time) among different datasets is very unlikely, object-field representations that record semantic and uncertainty information at different levels and not just at dataset level can give useful indications as to the classification/conceptualization process used by the producer at pixel and object level.

The model proposed is object driven. However, the old adage 'garbage in – garbage out' is apt. Thus, during the design of this model there has been a drive to improve the quality of a dataset by introducing semantic and uncertainty integrated multilevel objects in a traditional raster format to represent the semantic uncertainty inherent in collaborative GI modeling. The model is seen as a *processing system* which not only supports the visualization aspect but also includes objects that improve the output quality by allowing a group of people to attach, modify and share their semantics.

This paper proposes a framework of transferring geospatial conceptualizations and associated metadata that supports collaborative environments by realizing human-centred objects as if these were detached/real objects while field-based continuum representations are used. Thus, in a collaborative setting the framework seeks to make objects out of those continuous-smooth variations that afford associated human perceptions/beliefs and activities while transferring relevant semantic information explaining the human perceptions/beliefs and activities. These objects comprise an aggregate of locations connected together in space without existing necessarily as portions of geophysical reality. However, a user's certainty or doubt about his/her conceptual objects is also addressed. This certainty is addressed in order to establish a weighted attention (if necessary) in a collaborative voting system, for example.

Furthermore, the model is designed in such a way that object modifications are readily possible. The facets of a spatial problem can be added (with its semantic and uncertainty information) individually examined and accessed and if necessary removed based on, for example, [domain] object criteria. All of the required information are integrated in a single model opening the 'window' for the implementation of automatic processing algorithms that uses uncertainty, semantic and other quantified information. These semantic and uncertainty objects can have a number of purposes:

- To convince the collaborators of the view of the first user who is sharing his/her knowledge
- To transfer information that is not mappable
- To give the collaborators a context within which the collaborator conducted the analysis.

It is anticipated that this approach could work better than the traditional approaches by presenting the users with multimedia objects related to specific locations, sets of locations or the overall map while enabling the exploration and interrogation of these objects in an interactive human-controlled process.

In the future, since the model is intended to be implemented in a collaborative environment, human-computer interaction principles will be taken into account in the design of a spatial user interface that incorporates the model. In particular cognitive

overload issues (short, working and long memory), as semantic and uncertainty (additional) information may lead to additional overload problems. At the same time, semantic objects can provide the opportunity for analysing the viewed objects in terms of human observation based on symbolism confusion, which may bring new insights in the cognitive and semiotic approaches to understand maps as powerful and synthetic spatial representations. Furthermore, the analysis of a collaborator's perception and actions in the creation of objects may provide a *predictive model* of human behaviour and understanding giving an account of their beliefs as well as relating concepts corresponding to their environment in a direct and specific way which are indispensable for a collaborative environment.

References

1. Comber, A.J., Fisher, P., Wadsworth, R. (2004). "Integrating land cover data with different ontologies: identifying change from inconsistency". International Journal of Geographical Information Science 18 (7), 691-708
2. Couclelis, H.(1992). "People manipulate objects (but cultivate fields): Beyond the raster-vector debate in GIS". In: Frank, A.U., Campari, I., Formentini. (Eds.), Theories and Methods of Spatio-Temporal Reasoning in Geographic Space. Vol.639 of Lecture Notes in Computer Science. Springer, Berlin, pp. 65-77.
3. Cova, J. T. and M. F. Goodchild (2002). "Extending geographical representations to include fields of spatial objects." Internation Journal of Geographical Information Science 16(6): 509-532.
4. Dykes, J., MacEachren, A.M., Kraak, M.L. (2005). Exploring Geovisualization. In Exploring Geovisualization.(Eds, J. Dykes, A.M. MacEachren, M.L. Kraak), Amsterdam, Elsevier.
5. Fisher, P. F. (2003). "Multimedia Reporting of the Results of Natural Resource Surveys." Transactions in GIS 7(3): 309-324.
6. Guarino, N., (1998). Formal ontology and information systems. In 1st International Conference on Formal Ontology in Information Systems (Eds, N. Guarino), IOS Press:Trento, Italy.
7. Jung, S., Voser, A.V., Ehlers, M. (1998). "Hybrid spatial analysis operations as a foundation for integrated GIS." IAPRS 32(4):276-280, http://www.ifp.uni- stuttgart.de/ publications/commIV/jung87neu.pdf [23rd June, 2005]
8. Molenaar, M. (1998). An introduction to the theory of spatial object modelling, Taylor and Francis.
9. Peng, Z. P. (2005). "A proposed framework for feature-level geospatial shating: a case study for transportation network data." International Journal of Geographical Information Science 19(4).
10. Smith, B. and M. M. Mark (2002). "Do mountains exist? Towards an ontology of landforms." Environment and Planning B: Planning and Design 30: 411-427.
11. Tremblay, J. P. and Sorenson, P. G., 1985. An Introduction to Data Structures with Applications, Auckland: McGraw-Hill.
12. Winter, S.(1998). Bridging vector and raster representations in GIS. In: Laurini, R., Makki, K., Pssinou, . (Ed.), Proceedings of the 6th International Symposium on Advances in Geographic Information Systems, ACM Press, Washington, D.C., pp. 57-62.
13. Winter, S., Frank, A. U. (2000). "Topology in raster and vector representation." GeoInformatica 4(1):35-65

Semantics of Collinearity Among Regions

Roland Billen[1] and Eliseo Clementini[2]

[1] Dept. of Geography and Geomatics, University of Glasgow,
Glasgow G12 8QQ, Scotland, UK
rbillen@geog.gla.ac.uk
[2] Dept. of Electrical Engineering, University of L'Aquila,
I-67040 Poggio di Roio (AQ), Italy
eliseo@ing.univaq.it

Abstract. Collinearity is a basic arrangement of regions in the plane. We investigate the semantics of collinearity in various possible meanings for three regions and we combine these concepts to obtain definitions for four and more regions. The aim of the paper is to support the formalization of projective properties for modelling geographic information and qualitative spatial reasoning. Exploring the semantics of collinearity will enable us to shed light on elementary projective properties from which all the others can be inferred. Collinearity is also used to find a qualitative classification of the arrangement of many regions in the plane.

1 Introduction

Enriching the semantics of geographic information is a challenge that cannot avoid coping with the meaning of basic geometric concepts. Referring to the subdivision of geometric properties into topological, projective, and metric [3], we undertake the study of a basic projective property, which is collinearity among spatial objects. Collinearity is an elementary geometric notion among points and it is a very interesting issue how to extend this concept to two-dimensional regions.

The importance of semantics of collinearity relies on the fact that modelling all projective properties of spatial data can be done as a direct extension of such a property [1]. Notably, a model for representing the projective relations among spatial objects, the 5-intersection model, has been derived from the concept of collinearity among regions [2]. Most work on projective relations deals with point abstractions of spatial features and limited work has been devoted to extended objects [6, 7, 10]. In [4], the authors use spheres surrounding the objects to take into account the shape of objects in relative orientation. Early work on projective relations such as "between" was developed by [5].

Some interesting studies from theories of perception give support to our claim that collinearity is a basic property from which other properties may be derived [8]. "Emergent features" are visual properties possessed by configurations that are not present in the component elements of those same configurations. Examples of emergent features made up of two objects are proximity and orientation, emergent features made up of three objects are collinearity and symmetry, emergent features of four objects are surroundedness. Adding more objects, there are no further emergent features [9].

In Section 2, we describe the semantics of collinearity among three regions as an extension of the collinearity among points and we discuss the relevant formal properties. In section 3, we extend the concept of collinearity to four and more regions by taking two alternative ways, called step-wise collinearity and n-ary

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1066 – 1076, 2005.
© Springer-Verlag Berlin Heidelberg 2005

collinearity. In Section 4, we show that the collinearity of three regions is a concept that can be used to build a qualitative description about the arrangement of various regions in the plane. In Section 5, we draw short conclusions.

2 Semantics of Collinearity Among Three Regions

Collinearity among points is an elementary concept of projective geometry. At least three points are needed to define collinearity, and therefore it is intrinsically a ternary relation. Three points x,y,z are said to be *collinear* if they lie on the same line; we write $coll(x,y,z)$ in the rest of the paper.

Two groups of properties of the *collinear* relation that are important to discuss for extending the concept to regions are symmetry and transitivity. By symmetry, we mean that we can exchange the order of arguments in the relation. Symmetry can be expressed by the following equations:

$$\forall xyz \in R^2,\ coll(x,y,z) \Rightarrow coll(x,z,y),\ coll(x,y,z) \Rightarrow coll(z,x,y).$$

By transitivity, we mean that given four points and two *collinear* relations holding among them, we can infer collinearity for any triplet of points out of that set of four points; we write:

$$\forall xyzt \in R^2,\ coll(x,y,z) \wedge coll(y,z,t) \Rightarrow coll(x,z,t).$$

The extension of the collinearity relation among points to regions is rather complex. Indeed, its generalisation leads to different definitions of collinearity among regions. Basically, the collinearity relation is applied to points belonging to the three regions, but differences occur when one considers all the points of a region or just some of them. By different combinations of universal and existential quantifiers, we obtain eight different definitions that we call *collinear_1*, *collinear_2*, etc.; given three simple regions $A,B,C \in R^2$:

1. $coll_1(A,B,C) \equiv_{def} \exists x \in A\ [\exists y \in B\ [\exists z \in C\ [coll(x,y,z)]]]$;
2. $coll_2(A,B,C) \equiv_{def} \forall x \in A\ [\exists y \in B\ [\exists z \in C\ [coll(x,y,z)]]]$;
3. $coll_3(A,B,C) \equiv_{def} \exists x \in A\ [\forall y \in B\ [\exists z \in C\ [coll(x,y,z)]]]$;
4. $coll_4(A,B,C) \equiv_{def} \exists x \in A\ [\exists y \in B\ [\forall z \in C\ [coll(x,y,z)]]]$;
5. $coll_5(A,B,C) \equiv_{def} \forall x \in A\ [\forall y \in B\ [\exists z \in C\ [coll(x,y,z)]]]$;
6. $coll_6(A,B,C) \equiv_{def} \forall x \in A\ [\exists y \in B\ [\forall z \in C\ [coll(x,y,z)]]]$;
7. $coll_7(A,B,C) \equiv_{def} \exists x \in A\ [\forall y \in B\ [\forall z \in C\ [coll(x,y,z)]]]$;
8. $coll_8(A,B,C) \equiv_{def} \forall x \in A\ [\forall y \in B\ [\forall z \in C\ [coll(x,y,z)]]]$.

In the above definitions, the ordering of the quantifiers is the same as the ordering of variables in $coll(x,y,z)$: we could consider other variants of these relations by exchanging the ordering of variables in $coll(x,y,z)$, but they would not be significant since the collinear relation among points is symmetric. All the collinearity relations among regions can be hierarchically structured (Fig. 1); *Collinear_1* is at the top of the structure, all of the other cases are specialisations of it. This relation allows for

weaker constraints on the arrangement of three regions. At the opposite, *collinear_8* represent the strongest notion of collinearity. Between them, we have the other relations ruled by different levels of dependency, such as:

$$coll_2(A,B,C) \Rightarrow coll_1(A,B,C)$$
$$coll_5(A,B,C) \Rightarrow coll_2(A,B,C) \wedge coll_3(A,B,C)$$

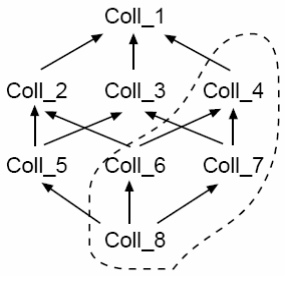


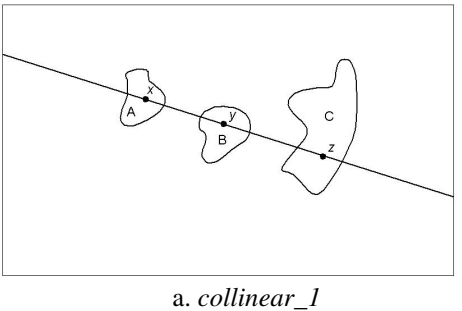
Fig. 1. Collinear relations' structure

Relations *collinear_4*, *collinear_6*, *collinear_7*, and *collinear_8* can be only applied to degenerate cases where at least one region is a segment or a point. Only *collinear_8* is transitive, and only *collinear_1* and *collinear_ 8* are symmetric.

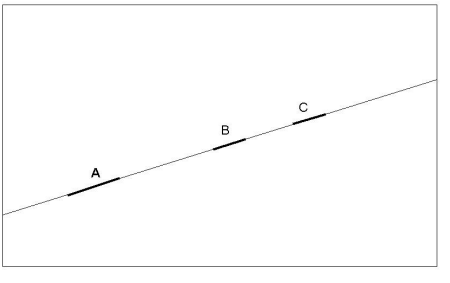
Collinear_1 relation

This relation is the easiest one to understand and surely the most useful. In short, three regions *A*, *B* and *C* are collinear if it exists at least one common line intersecting them (Fig. 2.a). This relation is *symmetric*, which means that the relation remains true under any permutation of the arguments. This can be expressed by the following relationships:

$$coll_1(A,B,C) \Rightarrow coll_1(A,C,B)\ ;$$
$$coll_1(A,B,C) \Rightarrow coll_1(B,A,C)\ .$$



a. *collinear_1*



b. *collinear_8*

Fig. 2. Symmetric collinear relations

Collinear_8 relation

The *collinear_8* relation is an extreme case which works only for degenerate regions (points or lines). It implies that each 3-tuple of points of *A*, *B* and *C* are collinear. It is

only possible if the regions are collinear points or collinear lines (or a combination of both) (figure 2.b).

The remaining 6 relations are not symmetric, it means that a relation would not be necessarily maintained after permutation of the order of the regions in the relation. It is therefore necessary to consider a primary object, region *A*, for whom the relation stands and two reference objects, regions *B* and *C*.

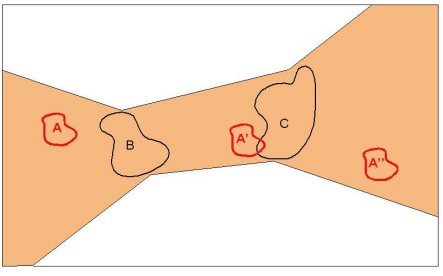


Fig. 3. *Collinear_2*

Collinear_2 relation

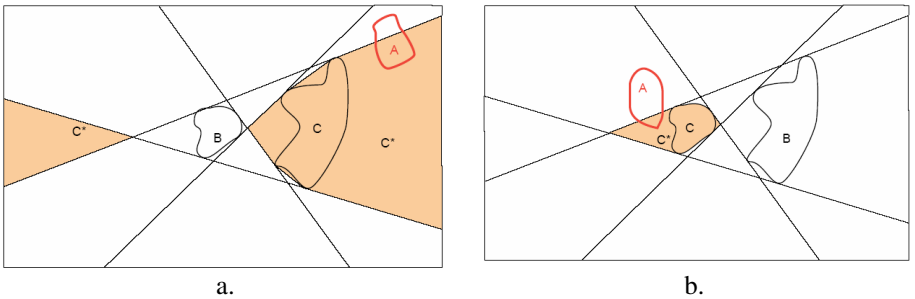
This relation means that every points of the primary object *A* have to be collinear with at least two points of *B* and *C*. It is said to be *partially symmetric*, which means that reference objects *B* and *C* can be exchanged:

$$coll_2(A,B,C) \Leftrightarrow coll_2(A,C,B)\ .$$

This relation has already been developed in previous work, and is the basis of the 5-intersection model [2]. At this stage, one can introduce the concept of *collinearity zone*: a *collinearity_2 zone* is the part of the plane where the relation *coll_2(A,B,C)* is true for any region *A* entirely contained in it. The *collinearity_2 zone* can be built using external and internal tangents of the regions *B* and *C* [2]. The concept is illustrated in Fig. 3.

Collinear_ 3 relation

Let us call *C** a zone including *C* and bounded by the external and the internal tangents like illustrated in Fig. 4. The relation *coll_3* (*A,B,C*) is true if region *A* is at least partially contained in *C**. Depending on the relative size and shape of *B* and *C*, the zone *C** can be bounded or unbounded.

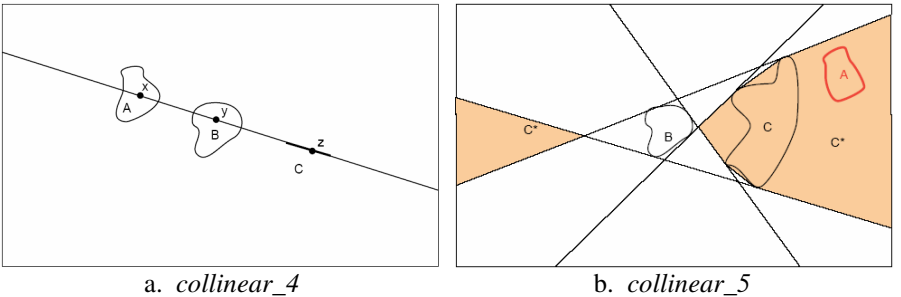


a. b.

Fig. 4. *Collinear_3.* The zone *C** can be unbounded and separated in two parts (a) or bounded (b)

Collinear_ 4 relation
The *collinear_4* relation is illustrated in Fig. 5.a. This relation is verified only if the object *C* is a segment contained in a line that intersects both *A* and *B*.

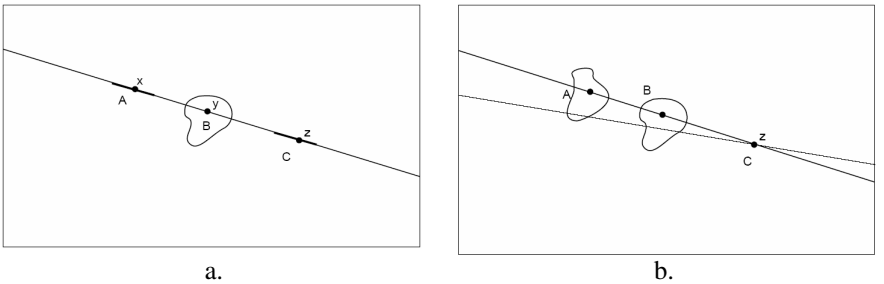
Collinear_5 relation
Considering the zone *C**, as defined for *collinear_3*, the relation *coll_5(A,B,C)* is true if region *A* is entirely contained in *C** (Fig. 5.b).



a. *collinear_4* b. *collinear_5*

Fig. 5. *Collinear_4* and *collinear_5*

Collinear_ 6 relation
The *collinear_6* relation is illustrated in Fig. 6. This relation is verified only in two cases: if objects *A* and *C* are segments belonging to a common line that is also intersecting *B* (Fig. 6.a); if *C* is a point and all lines passing through *A* and *C* intersect *B* (Fig. 6.b).

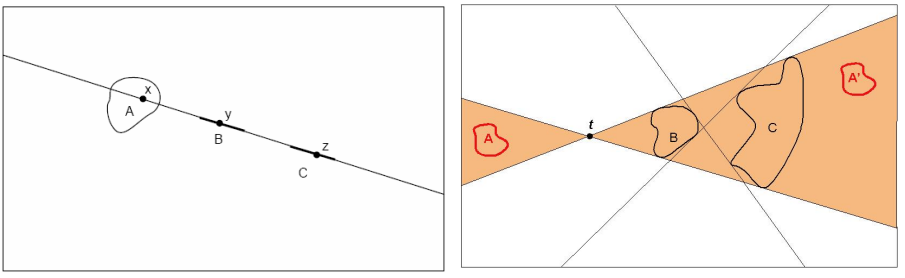


a. b.

Fig. 6. *Collinear_6*

Collinear 7 relation
Collinear_7 relation is partially symmetric. This relation is true only if *B* and *C* are two segments contained in a common line that intersects *A* (Fig. 7.a).

One may argue that other definitions of collinearity could be proposed. For example, one could define a *collinearity zone* bounded by the external tangents of *B* and *C* as it is illustrated in Fig. 7.b. However, such a relation can be expressed using



a. *collinear_7* b. another collinear relation

Fig. 7. *Collinear_7* and another collinear relation

previous definitions; in this particular case, it is a combination of *collinear_5* and the convex hull of regions *B* and *C*:

$$coll_9(A,B,C) \Leftarrow coll_5(A,B,C) \vee coll_5(A,C,B) \vee (A \subset CH(B \cup C)) .$$

Collinear_1 and *collinear_2* seem to be the most suitable definitions for expressing collinearity among three regions; *collinear_1* is the only usable symmetric case, and *collinear_2* is the most intuitive among the non-symmetric cases. Furthermore, when considering reference objects for *collinear_1*, one can see that the two relations share some common geometries; *coll_1 (A,B,C)* is true for any region *A* at least partially contained in the *collinearity_2 zone*.

As a twin concept of collinearity, one can define the fact of "being aside" by the negation of being collinear. We adopt the following definition:

$$aside(A,B,C) \Leftarrow \neg coll_1(A,B,C) .$$

The part of the plane where a region *A* satisfies the *aside* relation corresponds to the complement of the *collinearity_2 zone* and may be called the *aside zone*.

3 Semantics of Collinearity of Four and More Regions

We have two ways of extending the concept of collinearity to four and more regions. In the first way, given n regions *A, B, C, D, E, …*, we apply the already discussed definition of collinearity to groups of three regions taken in a given sequence: we call it *step-wise collinearity*. In the second way, we redefine collinearity directly from the concept of collinearity among n points: we call it *n-ary collinearity*.

Note that in the case of points, the two ways are equivalent since the transitive property holds for the relation collinear among three points; *n* points are collinear if there exists a single line that contains them all. If n points *x, y, z, t, …*, are collinear, we write *coll (x,y,z,t,…)*.

3.1 Step-Wise Collinearity

Step-wise collinearity can be defined for four and more regions for all the eight kinds of collinearity of Section 2. Given a sequence of regions *A*, *B*, *C*, *D*, *E*, … we have the following definitions for relations *collinear_1* and *collinear_2*:

$$sw_coll_1(A,B,C,D,E,...) \Leftarrow coll_1(A,B,C) \wedge coll_1(B,C,D) \wedge coll_1(C,D,E) \wedge ...$$

$$sw_coll_2(A,B,C,D,E,...) \Leftarrow coll_2(A,B,C) \wedge coll_2(B,C,D) \wedge coll_2(C,D,E) \wedge ...$$

The step-wise collinearity for the other kinds of collinearity is defined analogously, but it doesn't lead to interesting results. The relation *sw_collinear_1* as defined above leads to a too much weak form of collinearity among n regions. The relation *sw_collinear_2* is more interesting (see Fig. 8): it imposes a "local" collinearity for each triplet, but the global arrangement may be curvilinear and could end up also in a circular one by adding more regions to the sequence.

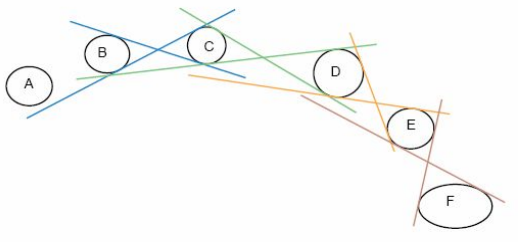


Fig. 8. *sw_collinear_2*

3.2 n-ary Collinearity

A more sophisticated way of defining collinearity for four and more regions is to pick up different combinations of existential and universal quantifiers for points of every region. In this way, we could theoretically find quite a number of definitions, but we restrict ourselves to the following two:

$$coll_1(A,B,C,D,E,...) \Leftarrow \exists x \in A, \exists y \in B, \exists z \in C, \exists t \in D, \exists u \in E, \ldots, \quad \text{such that } coll(x,y,z,t,u,...)$$

$$coll_2(A,B,C,D,E,...) \Leftarrow \forall x \in A, \exists y \in B, \exists z \in C, \exists t \in D, \exists u \in E, \ldots, \quad \text{such that } coll(x,y,z,t,u,...)$$

The relation *collinear_1* among many regions is symmetric in the same sense of relation *collinear_1* for three regions: the order in which the regions are considered does not affect the relation. An illustration of *collinear_1* is given in Fig. 9. The relation *aside* defined as the negation of being *collinear_1* stands for more than three objects.

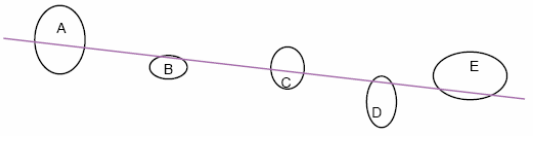


Fig. 9. n-ary *collinear_1*

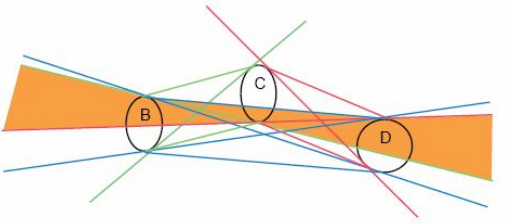


Fig. 10. n-ary *collinear_2* (with 3 reference objects)

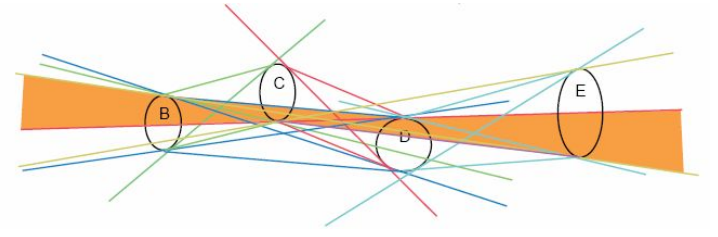


Fig. 11. n-ary *collinear_2* (with 4 reference objects)

The relation *collinear_2* among many regions is partially symmetric: we distinguish a primary object *A* and reference objects *B*, *C*, *D*, *E*, … but the reference objects only can be exchanged among themselves without affecting the relation *collinear_2*. It is possible to define a *collinearity zone* made up by the reference regions, which is the lieu of points a primary object can occupy to have the relation *collinear_2* satisfied. Geometrically, the *collinearity zone* can be obtained with the set intersection of all *collinearity zones* of all triplets of reference regions. This is illustrated in Fig. 10 for the *collinearity zone* formed by the regions *B*, *C*, and *D*, and in Fig. 11, where another region *E* is added to them. It is interesting to note that the *collinearity zone* tends to be narrower, when more reference regions are added. Therefore, the concept of collinearity we obtain put more constraints on the global arrangement of regions if they grow in number. The relation *collinear_2* implies the relation *collinear_1*, as it was for relations among three regions.

4 Categorizing Configurations Using Relation Collinear_1

Collinearity is a high-level primitive concept that can be used to formulate a qualitative description of the configuration of many regions in the plane. Such a description highlights what the relative position of regions is alike and can give information on the global arrangement of regions.

We consider the primitive relation *collinear_1* and its negation *aside*. For three regions, we can distinguish between two configurations: in one configuration, the three regions are *collinear_1* (Fig. 12.a), while in the other configuration the three regions are *aside* (Fig. 12.b). For four regions, the relation *collinear_1* can be checked

on various combinations of three regions obtaining a range of five different cases. The two extremes of this range are made up by the cases where all possible triplets of regions are *aside* (Fig. 12.c) and where all of them are *collinear_1* (Fig. 12.g). Then, there are intermediate cases where three triplets are *aside* and one of them is *collinear_1* (Fig. 12.d), two triplets are *aside* and two of them are *collinear_1* (Fig. 12.e) and one triplet is *aside* and three of them are *collinear_1* (Fig. 12.f).

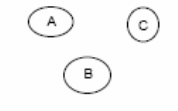
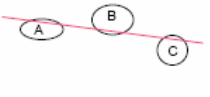
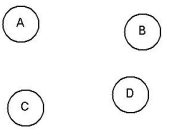
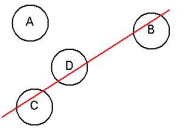
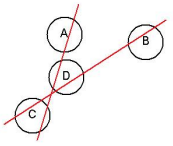
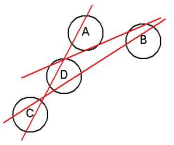
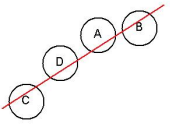
			
a. aside(A,B,C)	b. coll_1(A,B,C)	c. aside(A,B,C) ∧ aside(B,C,D) ∧ aside(C,D,A) ∧ aside(A,B,D)	d. aside(A,B,C) ∧ coll_1(B,C,D) ∧ aside(C,D,A) ∧ aside(A,B,D)
			
e. aside(A,B,C) ∧ coll_1(B,C,D) ∧ coll_1(C,D,A) ∧ aside(A,B,D)	f. aside(A,B,C) ∧ coll_1(B,C,D) ∧ coll_1(C,D,A) ∧ coll_1(A,B,D)	g. coll_1(A,B,C) ∧ coll_1(B,C,D) ∧ coll_1(C,D,A) ∧ coll_1(A,B,D)	

Fig. 12. Categorizing configurations using primitive *collinear_1*

Extending the counting of configurations to n regions in general, we can say that the number of different triplets corresponds to $k=\binom{n}{3}$. The number of possible relations *collinear_1* or *aside* for these triplets is 2^k: by grouping them in such a way the number of triplets being *collinear_1* is the same, we can distinguish k+1 different configurations. Such a number can be used as a rough measure of the "amount of collinearity" of a certain configuration of regions, ranging from 0, where all regions are *aside*, to k, where all regions are *collinear_1*. An intermediate value would state an intermediate degree of collinearity inside the configuration.

5 Conclusions

While projective geometry provides a precise description of an arrangement of points in the plane, it is not evident how a similar qualitative description can be obtained for objects having a two-dimensional extension. A fundamental step in this process is exploring the meaning of three regions being collinear. Such a concept must be

intrinsically an approximation since three regions cannot be collinear in a strict sense having different sizes and shapes. We found different ways of extending the concept from points to regions leading to a hierarchy of collinear relations, from the most permissive to the most stringent, and we commented their formal properties.

As a second step, we explored ways of combining the collinear relations among three regions to describe collinearity among many regions. This step is not obvious, since collinearity among regions loses the transitivity property, which in the case of points makes possible the extension from three to many points. We distinguished two categories of collinearity for many regions: the step-wise collinearity and the n-ary collinearity.

Finally, we used the concept of collinearity as a basic criterion to build a qualitative description of the arrangements of many regions in the plane: the number of collinear relations among triplets of regions expresses a measure of the collinearity of all the regions. In essence, a low number of collinear relations indicates an "encircling" arrangement of regions, while a high number of collinear relations indicates a tendency that the regions are located along the same line.

Acknowledgements

This work was supported by Italian M.I.U.R. under project "Representation and management of spatial and geographic data on the Web" and by the International Exchange Programme of the Royal Society of Edinburgh.

References

[1] R. Billen and E. Clementini, "Étude des caractéristiques projectives des objets spatiaux et de leurs relations," *Revue Internationale de Géomatique*, vol. 14, pp. 145-165, 2004.

[2] R. Billen and E. Clementini, "A model for ternary projective relations between regions," EDBT2004 - 9th International Conference on Extending DataBase Technology, Heraklion-Crete, Greece, pp. 310-328, 2004.

[3] E. Clementini and P. Di Felice, "Spatial Operators," *ACM SIGMOD Record*, vol. 29, pp. 31-38, 2000.

[4] V. Dugat, P. Gambarotto, and Y. Larvor, "Qualitative Theory of Shape and Orientation," Proc. of the 16th Int. Joint Conference on Artificial Intelligence (IJCAI'99), Stockolm, Sweden, pp. 45-53, 1999.

[5] K.-P. Gapp, "From Vision to Language: A Cognitive Approach to the Computation of Spatial Relations in 3D Space," Proc. of the First European Conference on Cognitive Science in Industry, Luxembourg, pp. 339-357, 1994.

[6] R. Goyal and M. J. Egenhofer, "Cardinal directions between extended spatial objects," http://www.spatial.maine.edu/~max/RJ36.html, 2000.

[7] L. Kulik, C. Eschenbach, C. Habel, and H. R. Schmidtke, "A graded approach to directions between extended objects," Proc. of the 2nd Int. Conf. on Geographic Information Science, Boulder, CO, pp. 119-131, 2002.

[8] J. R. Pomerantz, M. C. Portillo, S. W. Jewell, and A. Agrawal, "The Genesis of Perceptual Organization: Basic Emergent Features in Vision," 44th annual meeting of the Psychonomic Society, Vancouver, British Columbia, 2003.

[9] M. C. Portillo, J. R. Pomerantz, and S. Zimmerman, "Evaluating grouping via emergent features: A systematic approach," Fifth Annual Meeting of the Vision Science Society, Sarasota, FL, 2005.

[10] C. Vorwerg, G. Socher, T. Fuhr, G. Sagerer, and G. Rickheit, "Projective relations for 3D space: Computational model, application, and psychological evaluation," Proc. of the 14th Nat. Conf. on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conf., AAAI 97, IAAI 97, Providence, RI, pp. 159-164, 1997.

Automatic Acquisition of Fuzzy Footprints

Steven Schockaert, Martine De Cock, and Etienne E. Kerre

Department of Applied Mathematics and Computer Science,
Fuzziness and Uncertainty Modelling Research Unit,
Ghent University, Krijgslaan 281 (S9), B-9000 Gent, Belgium
{Steven.Schockaert, Martine.DeCock, Etienne.Kerre}@UGent.be
http://www.fuzzy.ugent.be

Abstract. Gazetteer services are an important component in a wide variety of systems, including geographic search engines and question answering systems. Unfortunately, the footprints provided by gazetteers are often limited to a bounding box or even a centroid. Moreover, for a lot of non–political regions, detailed footprints are nonexistent since these regions tend to have gradual, rather than crisp, boundaries. In this paper we propose an automatic method to approximate the footprints of crisp, as well as imprecise, regions using statements on the web as a starting point. Due to the vague nature of some of these statements, the resulting footprints are represented as fuzzy sets.

1 Introduction

Information on the web is often only relevant w.r.t. a particular geographic context. To this end, geospatial search engines [5,7] try to enhance the functionality of search engines by georeferencing web pages, i.e. by automatically assigning a geographic location to web pages. Consider for example the web page of a pizza restaurant in Gent. A geographic search engine would, for example, only return the web page of this restaurant if the user is located in East Flanders[1]. Geographic question answering systems [10] go even further, as they are able to respond to natural language questions and requests from users such as "What are the neighbouring countries of Belgium?". Clearly, these systems have to make use of some kind of digital gazetteer to obtain the necessary background knowledge. To respond appropriately to a request like "Show me a list of pizza restaurants in the Ardennes."[2] a suitable footprint of the Ardennes is needed. However for reasons discussed in [4], the footprints provided by gazetteers are often restricted to a bounding box, or even a point (expressed by its latitude and longitude coordinates). For imprecise regions such as the Ardennes, providing a bounding box is not even feasible since this kind of regions is not characterized by a clearly defined boundary.

A promising solution is to construct the footprint of a particular region in an automatic way. In [1] a method based on Voronoi diagrams for approximating

[1] Gent is the capital of the province of East Flanders.
[2] The Ardennes is a region in the southern part of Belgium.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1077–1086, 2005.
© Springer-Verlag Berlin Heidelberg 2005

the footprint of a given region is proposed. Such a footprint is constructed from a set of points which are known to lie inside the region and a set of points which are known to lie outside the region. These sets of points are assumed to be correct and a priori available, e.g. provided by the user, hence this method is not fully automatic. In [9], it is suggested to represent regions with indeterminate boundaries by an upper and a lower approximation. Upper and lower approximations are constructed based on a set of points or regions which are known to be a part of the region under consideration, and a set of regions which are known to include the region under consideration. Again these sets are assumed to be correct and available a priori. A fully automatic algorithm is introduced in [2] where statements on the web such as "... in Luxembourg and other Ardennes towns ..." are used to obtain a set of points which are assumed to lie inside the region under consideration. To obtain a footprint from this set of points, the algorithm from [1] is slightly modified to cope with the noisiness of data from the web. Finally, in [8] kernel density surfaces are used to represent imprecise regions. However it is unclear what meaning should be attached to the weights corresponding to each point, as these weights seem to reflect the popularity (e.g. expressed as the number of occurrences of the corresponding city on the web) of the corresponding cities rather than some kind of vague representation of a region.

In this paper we introduce a new method to automatically construct a footprint for, possibly imprecise, regions by extracting relevant statements from the web. In contrast to existing approaches [2,8] we do not only search for places that lie in the region under consideration, but also for regions that include this region, and for regions that are bordering on this region. Moreover, we use statements on the web such as "x is in the south-western corner of $\mathcal{R}$" to constrain the possible cities that could lie in the region $\mathcal{R}$. Due to the vagueness of this type of constraints, we propose using possibility distributions to this end. Information on the web can be inaccurate, outdated or even simply wrong. Hence, enforcing every constraint that is found on the web can result in an inconsistent solution (e.g. the only possible footprint is the empty region). Therefore, we apply ideas from the theory of fuzzy belief revision to (partially) discard certain constraints in the face of inconsistencies. The resulting footprint of the region is represented as a fuzzy set, which we call a fuzzy footprint in this context.

2 Obtaining Data from the Web

2.1 Acquiring Place Names Through Regular Expressions

Assume that we want to approximate the extent of a (possibly imprecise) region $\mathcal{R}$. The first step of our algorithm consists of searching the web for relevant statements and extracting useful data from these statements. In order to find relevant statements we send a number of queries to Altavista[3] such as "$\mathcal{R}$", "$\mathcal{R}$ is located in", "in $\mathcal{R}$ such as", ... and analyse the snippets that are returned.

[3] `http://www.altavista.com`

Table 1. Regular expressions

Abbreviations
`<direction> = (heart | centre | ...| north-west | north-western)`
`<place> = (village | villages | town | towns | city | cities)`
`<area> = (region | province | state | territory)`
`<names> = <name> (, <name>)* (and <name>)?`
`<name> = [A-Z][a-z]+ ([A-Z][a-z]+)?`
`<dir-part> = the? <direction> (part | corner)? of?`
`<region-part> = the? ((<area> of)? R | R <area>?)`
`<dir-reg> = <dir-part>? <region-part>`

Regular expressions to find points inside $\mathcal{R}$	
1.	`(located|situated) in <dir-reg> (the <place> of)? <names>`
2.	`<names> (is|are) (a? <place>)? in <dir-reg>`
3.	`<names> (is|are) (located|situated) in <dir-reg>`
4.	`<names>, (located|situated) in <dir-reg>`
5.	`<names> and (a lot of)? other <place> in <dir-reg>`
6.	`<place> in <dir-reg> (are|such as|like|including) <names>`

Regular expressions to find regions bordering on $\mathcal{R}$	
7.	`<name> <area>? which borders the? (<area> of)? R`
8.	`R <area>? which borders the? (<area> of)? <name>`
9.	`<name> <area>? bordering (on|with)? the? (<area> of)? R`
10.	`R <area>? bordering (on|with)? the? (<area> of)? <name>`

Note that for reasons of efficiency we only analyse the snippets, and do not fetch the corresponding full documents. From these snippets we want to obtain:

1. A set P of points that are assumed to lie in $\mathcal{R}$.
2. The country $\mathcal{S}$ that is assumed to include $\mathcal{R}$[4].
3. A set B of regions that are assumed to border on $\mathcal{R}$.
4. A set C_P of constraints w.r.t. the positioning of some of the points in P (e.g. q is in the north of $\mathcal{R}$).
5. A set C_S of constraints w.r.t. the positioning of $\mathcal{R}$ in $\mathcal{S}$ (e.g. $\mathcal{R}$ is in the north of $\mathcal{S}$).

To this end we adopt a pattern-based approach using the regular expressions in Table 1. The regular expressions 1–6 can be used to find places in $\mathcal{R}$ and some corresponding constraints, i.e. to construct P and C_P. The regular expressions that are used to construct $\mathcal{S}$ and C_S (not shown) are entirely analogous. Finally, the regular expressions 7–10 can be used to find bordering regions, i.e. to construct B.

[4] If there are several possible countries found that may include $\mathcal{R}$, the algorithm could simply be repeated for each candidate, and the optimal solution could be selected afterwards. Furthermore, we could also consider the union of several (neighboring) countries to cope with regions whose extent spans more than one country.

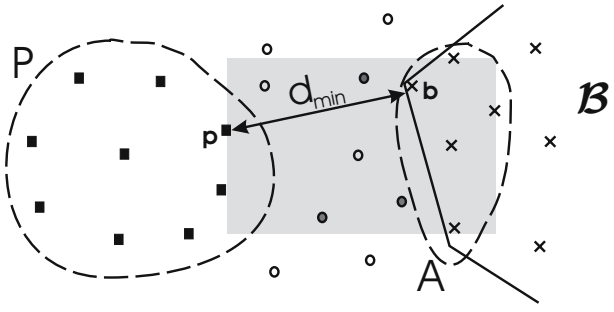


Fig. 1. Considering bordering region $\mathcal{B}$ to extend P. P is the set of points assumed to be in the region $\mathcal{R}$ to be approximated

2.2 Grounding the Place Names

The next step consists of grounding the geographic names that have been found. To accomplish this we use the Alexandria Digital Library (ADL) gazetteer[5], an online gazetteer service which can be accessed by using an XML- and HTTP-based protocol. To ground the place names in P, we first discard all names that are not located in $\mathcal{S}$. To disambiguate the remaining names, we choose the interpretations of the names in such a way that the area of the convex hull of the corresponding locations is minimal. This is a well-known heuristic which is, for example, described in more detail in [6]. To ground the bordering regions in B, we only consider administrative regions which are a part of $\mathcal{S}$, or border on $\mathcal{S}$. Since the footprint of most administrative regions provided by the ADL gazetteer is a point (the centroid of the region), for each bordering region $\mathcal{B}$ we construct a more accurate footprint by determining the convex hull of the places that are known to be in $\mathcal{B}$ by the ADL gazetteer. Consequently, for each bordering region $\mathcal{B}$ we calculate the minimal distance d_{min} from a point p in P to a place b in $\mathcal{B}$. Let A be the set of places in $\mathcal{B}$ for which the distance to p is less than $\lambda \cdot d_{min}$ where $\lambda \geq 1$. We now make the assumption that all places that lie within the minimal bounding box of $A \cup \{p\}$ and are not known to lie in the bordering region $\mathcal{B}$, lie in $\mathcal{R}$. Therefore, we add the most northern, most southern, most western and most eastern of these places to P. Note that one of these places will be p. Adding all places is not desirable, as this would influence too much the median of P and the average distance between the places in P. This process is illustrated in Figure 1.

3 Constructing Solutions

So far we have used information about the country to which $\mathcal{R}$ belongs as well as about bordering regions of $\mathcal{R}$ to update the set of points P that are assumed to lie in $\mathcal{R}$. In this section we use the remaining data retrieved from the web

[5] `http://www.alexandria.ucsb.edu/gazetteer/`

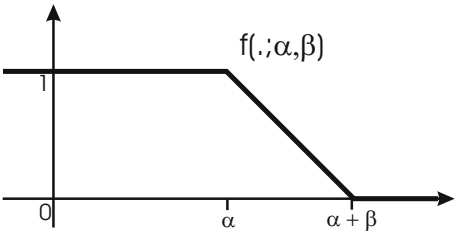


Fig. 2. The function $f(.;\alpha,\beta)$

to further enhance P, namely the sets of constraints C_P and C_S. Our aim is to construct a fuzzy set in P, i.e. a $P-[0,1]$ mapping F called a fuzzy footprint of $\mathcal{R}$. For each p in P, $F(p)$ is interpreted as the degree to which the point p belongs to $\mathcal{R}$. This membership degree is computed based on the constraints retrieved from the web. Each constraint is represented by a $P-[0,1]$ mapping c, called a possibility distribution in P. For every point p in P, $c(p)$ is the possibility that p lies in $\mathcal{R}$, taking into account the constraint modelled by c. For more information about fuzzy set theory and possibility theory, we refer to [12].

3.1 Modelling the Constraints

First, consider constraints of the form "q is in the north of $\mathcal{R}$", where q is a place in P. If p is south of q, the possibility that p lies in $\mathcal{R}$ remains 1. However, the further north of q that point p is situated, the less possible it becomes that p lies in $\mathcal{R}$. To construct the corresponding possibility distribution we use the function f depicted in Figure 2 as well as the average difference in y–coordinates between the points in P, i.e.

$$\Delta_y^{avg} = \frac{1}{|P|^2} \sum_{p \in P} \sum_{q \in P} |p_y - q_y| \tag{1}$$

The constraint "q is in the north of $\mathcal{R}$" is then modelled by the possibility distribution c_q^N, defined for each p in P as

$$c_q^N(p) = f(p_y - q_y; \alpha_1 \Delta_y^{avg}, \beta_1 \Delta_y^{avg}) \tag{2}$$

where $\alpha_1 \geq 0$ and $\beta_1 \geq 0$ are constants. Hence if $p_y - q_y \leq \alpha_1 \Delta_y^{avg}$ then the possibility that p lies in $\mathcal{R}$ is 1; if $p_y - q_y \geq (\alpha_1 + \beta_1)\Delta_y^{avg}$, then the possibility that p lies in $\mathcal{R}$ is 0; in between there is a gradual transition. In the same way, we can express that q is in the south, east or west of $\mathcal{R}$. Constraints of the form "q is in the north-west of $\mathcal{R}$" are separated in "q is in the north of $\mathcal{R}$" and "q is in the west of $\mathcal{R}$". Constraints of the form "q is in the middle of $\mathcal{R}$" can be represented in a similar way.

Constraints of the form "$\mathcal{R}$ is in the north of $\mathcal{S}$" are easier to model, since the exact bounding box of $\mathcal{S}$ is known. A point p from P is consistent with the

constraint "$\mathcal{R}$ is in the north of $\mathcal{S}$" if it is in the northern half of this bounding box. Again, a fuzzy approach can be adopted where points that are slightly below the half are considered consistent to a certain degree. In the same way, we can express that $\mathcal{R}$ is in the south, east, west or centre of $\mathcal{S}$. Another type of constraints is induced by the elements of B. Each bordering region $\mathcal{B}$ in B induces a possibility distribution $c_{\mathcal{B}}$ on P, defined for each point p from P by $c_{\mathcal{B}}(p) = 0$ if p lies in $\mathcal{B}$ and $c_{\mathcal{B}}(p) = 1$ otherwise. In other words, if $\mathcal{B}$ is a bordering region of $\mathcal{R}$, $\mathcal{R}$ and $\mathcal{B}$ cannot overlap. In the following, let C_B be the set of all constraints induced by B.

Finally we impose an additional constraint c_h which is based on the heuristic that outliers in the set P are not likely to be correct. Let d be a distance metric on P (e.g. the Euclidean distance), we define the median m of P as $m = \arg\min_{p \in P} \sum_{q \in P} d(p, q)$. The possibility distribution c_h can be defined for each p in P by

$$c_h(p) = f(d(p, m); \alpha_2 d_{avg}, \beta_2 d_{avg}) \tag{3}$$

where $\alpha_2 \geq 0$ and $\beta_2 \geq 0$ are constants, and $d_{avg} = \frac{1}{|P|} \sum_{p \in P} d(m, p)$. In other words, the closer p is to the median m of P, the more possible it is that p lies in $\mathcal{R}$.

3.2 Resolving Inconsistencies

Let the set C be defined as $C = C_P \cup C_S \cup C_B \cup \{c_h\}$. Each of the possibility distributions in C restricts the possible places that could lie in $\mathcal{R}$. If each constraint were correct, we could represent the footprint of $\mathcal{R}$ as the fuzzy set F defined for p in P by

$$F(p) = \min_{c \in C} c(p) \tag{4}$$

This is a conservative approach in which the membership degree of p in $\mathcal{R}$ is determined by the constraint c that restricts the possibility of p lying in $\mathcal{R}$ the most. In practice however, C is likely to contain inconsistent information either because some websites contain erroneous information, because the use of regular expressions could lead to a wrong interpretation of a sentence, or because our interpretation of the constraints is too strict. As a consequence of these inconsistencies, F would not be a normalised fuzzy set, i.e. no point p would belong to F to degree 1, and could even be the empty set. To overcome this anomaly, we use a $C - [0, 1]$ mapping K such that for c in C, $K(c)$ expresses our belief that c is correct. Formally, K is a fuzzy set in C, i.e. a fuzzy set of constraints. The fuzzy footprint corresponding with K is the fuzzy set F_K in P defined for p in P by

$$F_K(p) = \min_{c \in C} I_W(K(c), c(p)) \tag{5}$$

using the fuzzy logical implicator[6] I_W defined for a and b in $[0, 1]$ by

$$I_W(a, b) = \min(1, 1 - a + b) \tag{6}$$

[6] Implicators are $[0, 1]^2 - [0, 1]$ mappings which generalize the notion of implication from binary logic to the unit interval.

Eq. (5) expresses that we only impose the constraints in C to the degree that we believe they are correct. Note that if $K(c) = 1$ for all c in C (i.e. we are confident that all constraints are correct), then $F_K = F$. On the other hand if $K(c) = 0$ for all c in C (i.e. we reject all constraints), then $F_K = P$. The belief degrees in the constraints are determined automatically in a stepwise manner that can give rise to more than one optimal fuzzy set of constraints. We use $L^{(i)}$ to denote the class of optimal fuzzy sets A obtained in step i of the construction process; each A contains the first i constraints to a certain degree. Let $C = \{c_1, c_2, \ldots, c_n\}$ and $L^{(0)} = \{\emptyset\}$, i.e. $L^{(0)}$ is a set containing the empty set. For $i = 1, \ldots, n$ we define

$$L^{(i)} = \{A + c_i | A \in L^{(i-1)}\} \cup \{A \oplus c_i | A \in L^{(i-1)}\} \tag{7}$$

where $+$ is an expansion operator and $\oplus$ is a revision operator. The idea behind expansion is to add the next constraint c_i to A only to the degree α that c_i is consistent with A, i.e. to the highest degree α for which the footprint corresponding with the resulting fuzzy set of constraints is normalised. The idea behind revision is to select a particular fuzzy subset[7] $\hat{A}$ of A such that the footprint corresponding with $\hat{A}$ augmented with constraint c_i to degree 1 is normalised. In other words, for each constraint c_i that is not fully consistent with A we choose either to (partially) reject c_i, or to (partially) reject the constraints in A. For more details on fuzzy revision and expansion operators we refer to [3,11].

If there are no inconsistencies, $L^{(n)}$ will contain only one fuzzy set K, hence F_K is the only possible footprint. However in the face of inconsistencies, $L^{(n)}$ will contain a number of possible alternatives $K_1, K_2, \ldots, K_m$. To rank the possible candidates, we assign each K_i a score $s(K_i)$ defined by

$$s(K_i) = \frac{area(cvx(F_{K_i}))}{\max_{j=1}^{m} area(cvx(F_{K_j}))} \cdot \frac{\sum_{c \in C} K_i(c)}{\max_{j=1}^{m} \sum_{c \in C} K_j(c)} \tag{8}$$

where for a fuzzy set B in $\mathbb{R}^2$

$$area(B) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} B(x, y) dx\, dy \tag{9}$$

provided the integral exists. The convex hull $cvx(B)$ of B is defined as the smallest convex fuzzy superset of B, i.e. every other convex fuzzy superset of B is also a fuzzy superset of $cvx(B)$, where a fuzzy set B in $\mathbb{R}^2$ is called convex if

$$\big(\forall \lambda \in [0,1]\big)\big(\forall (x,y) \in \mathbb{R}^2 \times \mathbb{R}^2\big)\big(B(\lambda x + (1-\lambda)y) \geq \min(B(x), B(y))\big) \tag{10}$$

The score $s(K_i)$ expresses that optimal footprints should satisfy as many constraints as possible, while the corresponding extent should remain as large as possible.

[7] Let A and B be fuzzy sets in a universe X; A is called a fuzzy subset of B, or likewise B is a fuzzy superset of A, if and only if $(\forall x \in X)(A(x) \leq B(x))$.

4 Experimental Results

Because the footprint of an imprecise region is inherently subjective, we will focus in this section on political regions, which are characterized by an exact, unambiguous boundary. To this end, we will compare the fuzzy sets that result from our algorithm with a gold standard. As the gold standard for a region $\mathcal{R}$, we have used the convex hull of the place names that are known to lie in $\mathcal{R}$ by the ADL gazetteer. We will denote the gold standard for $\mathcal{R}$ by $\mathcal{R}^*$. Note that this gold standard is not a perfect footprint, among others because the *part-of* relation in the ADL gazetteer is not complete. Let $\mathcal{A}$ be a fuzzy set in $\mathbb{R}^2$; to assess to what extent $\mathcal{A}$ is a good approximation of $\mathcal{R}^*$, we propose the following measures:

$$s_p(\mathcal{A}) = incl(\mathcal{A}, \mathcal{R}^*) \qquad s_r(\mathcal{A}) = incl(\mathcal{R}^*, \mathcal{A})$$

where for A and B fuzzy sets[8] in a universe X

$$incl(A, B) = \frac{\sum_{x \in X} \min(A(x), B(x))}{\sum_{x \in X} A(x)} \tag{11}$$

s_p expresses the degree to which $\mathcal{A}$ is included in $\mathcal{R}^*$, i.e. the degree to which the places that lie in $\mathcal{A}$ also lie in $\mathcal{R}^*$; hence s_p can be regarded as a measure of precision. On the other hand, s_r expresses the degree to which $\mathcal{A}$ includes $\mathcal{R}^*$ and can be regarded as a measure of recall.

As test data we took 81 political subregions of France, Italy, Canada, Australia and China ("countries, 1st order divisions" in the ADL gazetteer). Table 2 and Table 3 show the values of $s_p(F_K)$ and $s_r(F_K)$ that were obtained using several variants of our algorithm, where F_K is the footprint with the highest score (Eq. (8)) that was constructed. As parameter values, we used $\alpha_1 = 0.5$, $\beta_1 = 1$, $\alpha_2 = 1.5$, $\beta_2 = 5$ and $\lambda = 1.5$. For the first four columns we didn't consider bordering regions (neither to extend P as in Section 2.2 nor to construct a set of constraints C_B); for the column 'no' no constraints were imposed, for 'c_h' only c_h was imposed, for 'C_P' only the constraints in C_P were imposed, and finally for 'C_P, C_S, c_h' the constraints in $\{c_h\} \cup C_P \cup C_S$ were imposed. For the last four columns, bordering regions were used to extend P as in Section 2.2. For the column 'all' the constraints in $\{c_h\} \cup C_P \cup C_S \cup C_B$ were imposed. Obviously for popular regions we will find more relevant cities, constraints and bordering regions. Therefore, we split the regions into three groups: regions for which at least 30 possible cities were found (11 regions), regions for which less than 10 possible cities were found (38 regions), and the other regions (32 regions). For popular regions, imposing the constraints significantly increases precision. Furthermore considering bordering regions significantly improves recall, provided not all constraints are imposed. Unfortunately, considering bordering regions also decreases precision drastically. We believe that this is, at least partially, caused by the fact

[8] Note that $\mathcal{R}^*$ is in fact an ordinary set. However ordinary sets can be treated as special cases of fuzzy sets for which the membership degrees take only values in $\{0, 1\}$.

Table 2. Precision $s_p(F_K)$

	no bordering regions				bordering regions			
	no	c_h	C_P	C_P, C_S, c_h	no	c_h	C_P	all
All regions	0.26	0.43	0.43	0.47	0.16	0.30	0.35	0.42
$\|P\| \geq 30$	0.35	0.70	**0.85**	**0.83**	0.15	0.43	0.57	0.62
$10 \leq \|P\| < 30$	0.31	0.51	0.51	0.57	0.19	0.36	0.43	0.51
$\|P\| < 10$	0.20	0.28	0.23	0.28	0.14	0.22	0.22	0.28

Table 3. Recall $s_r(F_K)$

	no bordering regions				bordering regions			
	no	c_h	C_P	C_P, C_S, c_h	no	c_h	C_P	all
All regions	0.49	0.39	0.35	0.32	0.57	0.49	0.44	0.37
$\|P\| \geq 30$	0.85	0.59	0.38	0.33	**0.91**	0.70	0.55	0.39
$10 \leq \|P\| < 30$	0.68	0.56	0.50	0.48	0.75	0.66	0.57	0.52
$\|P\| < 10$	0.23	0.19	0.21	0.17	0.32	0.28	0.30	0.25

that the *part-of* relation in the ADL gazetteer is not complete. Therefore if recall is considered less important than precision, bordering regions should not be used. On the other hand, if recall is considered more important than precision bordering regions should be used, but not all constraints should be imposed, e.g. only c_h or only the constraints in C_P.

5 Conclusions

We have proposed a novel method to approximate the footprint of a (possibly imprecise) region by using statements on the web as a starting point. Existing approaches consider only statements that express that a particular city lies in the region of interest. We have extended this by also considering bordering regions and regions that are assumed to include the region of interest. Moreover, we have proposed to interpret vague restrictions such as "x is in the north-western corner of $\mathcal{R}$" and thus reducing the noise which is inevitably apparent when using data from the web. As a consequence, the resulting footprint is represented as a fuzzy set instead of, for example, a polygon. Inconsistencies between the constraints are resolved by using ideas from the theory of (fuzzy) belief revision. The experimental results show that imposing constraints can significantly improve precision, while considering bordering regions improves recall.

Acknowledgments

Steven Schockaert and Martine De Cock would like to thank the Fund for Scientific Research – Flanders for funding their research.

References

1. Alani, H., Jones, C.B., Tudhope, D.: Voronoi-based region approximation for geographical information retrieval with gazetteers. Int. J. Geographical Information Science **15** (2001) 287–306
2. Arampatzis, A., van Kreveld, M., Reinbacher, I., Jones, C.B., Vaid, S., Clough, P., Joho, H., Sanderson, M., Benkert, M., Wolff A.: Web-based delineation of imprecise regions. Proc. of the Workshop on Geographic Information Retrieval, SIGIR. `http://www.geo.unizh.ch/~rsp/gir/` (2004)
3. Booth, R., Richter, E.: On revising fuzzy belief bases. UAI International Conference on Uncertainty in Artificial Intelligence (2003) 81-88
4. Hill, L.L.: Core elements of digital gazetteers: placenames, categories, and footprints. Lecture Notes in Computer Science **1923** (2000) 280–290
5. Jones, C.B., Purves, R., Ruas, A., Sanderson, M., Sester, M., van Kreveld, M., Weibel R.: Spatial information retrieval and geographical ontologies: an overview of the SPIRIT project. Proc. of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (2002) 387–388
6. Leidner, J.L., Sinclair, G., Webber, B.: Grounding spatial named entities for information extraction and question answering. Proc. of the Workshop on the Analysis of Geographic References, NAACL-HLT (2003) 31–38
7. Markowetz, A., Brinkhoff, T., Seeger, B.: Geographic information retrieval. Proc. of the 3rd International Workshop on Web Dynamics. `http://www.dcs.bbk.ac.uk/webDyn3/webdyn3_proceedings.pdf` (2004)
8. Purves, R., Clough, P., Joho, H.: Identifying imprecise regions for geographic information retrieval using the web. Proc. of the GIS Research UK 13th Annual Conference (to appear)
9. Vögele, T., Schlieder, C., Visser, U.: Intuitive modelling of place name regions for spatial information retrieval. Lecture Notes in Computer Science **2825** (2003) 239–252
10. Waldinger, R., Appelt, D.E., Fry, J., Israel, D.J., Jarvis, P., Martin, D., Riehemann, S., Stickel, M.E. , Tyson, M., Hobbs, J., and Dungan, J.L.: Deductive question answering from multiple resources. In: Maybury, M. (ed.): New Directions in Question Answering. AAAI Press (2004) 253–262
11. Witte, R.: Fuzzy belief revision. 9th Int. Workshop on Non-Monotonic Reasoning (2002) 311–320
12. Zadeh, L.A.: Fuzzy sets as the basis for a theory of possibility. Fuzzy Sets and Systems **1** (1978) 3–28

The Double-Cross and the Generalization Concept as a Basis for Representing and Comparing Shapes of Polylines

Nico Van de Weghe[1], Guy De Tré[2], Bart Kuijpers[3], and Philippe De Maeyer[1]

[1] Department of Geography, Ghent University,
Krijgslaan 281 (S8), B-9000 Ghent, Belgium
[2] Computer Science Laboratory, Ghent University,
St.-Pietersnieuwstraat 41, B-9000 Ghent, Belgium
[3] Theoretical Computer Science Group, Hasselt University,
Agoralaan, Gebouw D, B-3590 Diepenbeek, Belgium

Abstract. Many shape recognition techniques have been presented in literature, most of them from a quantitative perspective. Research has shown that qualitative reasoning better reflects the way humans deal with spatial reality. The current qualitative techniques are based on break points resulting in difficulties in comparing analogous relative positions along polylines. The presented shape representation technique is a qualitative approach based on division points, resulting in shape matrices forming a shape data model and thus forming the basis for a cognitively relevant similarity measure for shape representation and shape comparison, both locally and globally.

1 Introduction

Specific domains of interest for shape retrieval include computer vision, image analysis and GIScience. Roughly speaking, there are two general approaches of shape retrieval: the quantitative approach and the qualitative approach. Traditionally, most attention has gone to the quantitative methods, e.g. [1-4]. Recently, the qualitative approach has gained more attention. What's more, cognitive studies provide evidence that qualitative models of shape representation tend to be much more expressive than its quantitative counterpart [5]. On the other hand, it was stated in [6] that shape is one of the most complex phenomena that has to be dealt with in the qualitative representation of space. There are two major qualitative approaches for representing two-dimensional shapes: the region-based and the boundary-based approach. The region-based approach uses global descriptors such as circularity, eccentricity and axis orientation to describe shapes. As a result, this approach can only discriminate shapes with large dissimilarities [7]. By use of a string of symbols, the boundary-based approach describes the type and position of localized features (such as vertices, extremes of curvature and changes in curvature) along the polyline representing the shape [6]. Among the boundary-based approaches are [6-14].

In this work, we present the Qualitative Trajectory Calculus for Shapes (QTC_S), as being a basis to represent and to compare shapes. In Section 2, two central concepts laying at the basis of QTC_S are presented: the *Double-Cross Concept* and the

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1087 – 1096, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Generalization Concept. Section 3 handles similarity based on QTC_S. In Section 4, advantages compared with existing boundary-based approaches are discussed. Finally, Section 5 handles some areas for further research.

2 Central Concepts of the Qualitative Trajectory Calculus for Shapes (QTC_S)

The Qualitative Trajectory Calculus for Shapes (QTC_S) is based on two central concepts. Based on the Double-Cross Calculus, the *Double-Cross Concept* is presented, as an expressive way of qualitatively representing a configuration of two vectors by means of a 4-tuple representing the orientation of both vectors with respect to each other. The *Generalization Concept* is an elegant way to overcome problems that are inherent on traditional boundary-based approaches.

2.1 Double-Cross Concept

In [15], a qualitative spatio-temporal representation for moving objects based on describing their relative trajectories, the so-called Qualitative Trajectory Calculus (QTC), has been developed. Important in this calculus, which is based on the Double-Cross Calculus [16,17], is that an object moving during an interval i (starting at time point t_1 and ending at time point t_2) is represented by means of a vector starting at t_1 and ending at t_2. If these vectors were not considered as representations of movements between two time points, but as vectors at a single moment in time, a spatial configuration of vectors can be analyzed, called the Qualitative Trajectory Calculus for Shapes (QTC_S).

In the qualitative approach, continuous information is being qualitatively discretized by landmarks resulting in discrete quantity spaces, such as {–, 0, +} consisting of the landmark value 0 and its neighboring open intervals $]-\infty,0[$ denoted by – and $]0,\infty[$ denoted by + [18]. The major idea in the qualitative approach is that a distinction is introduced only if it is relevant to the current research context [19,20]. Because it has been studied in experimental psychology that humans are poor at estimating angles and tend to use rectangular reference systems [16], one could say that a qualitative spatial calculus differentiating between parallelism and perpendicularity is as much as necessary. Hence, a vector pair ([*ab*],[*cd*]) is presented in QTC_S using the following conditions (C) (Fig. 1).

Assume: [*ab*] : the vector starting in vertex *a* and ending in vertex *b*
ab : the line through *a* and *b*
RL_{ac} : the directed reference line from *a* to *c*
$b\alpha ac$: the projection of *b* on *ac*

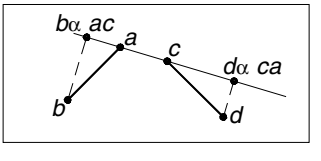


Fig. 1. Construction rules for QTC_S

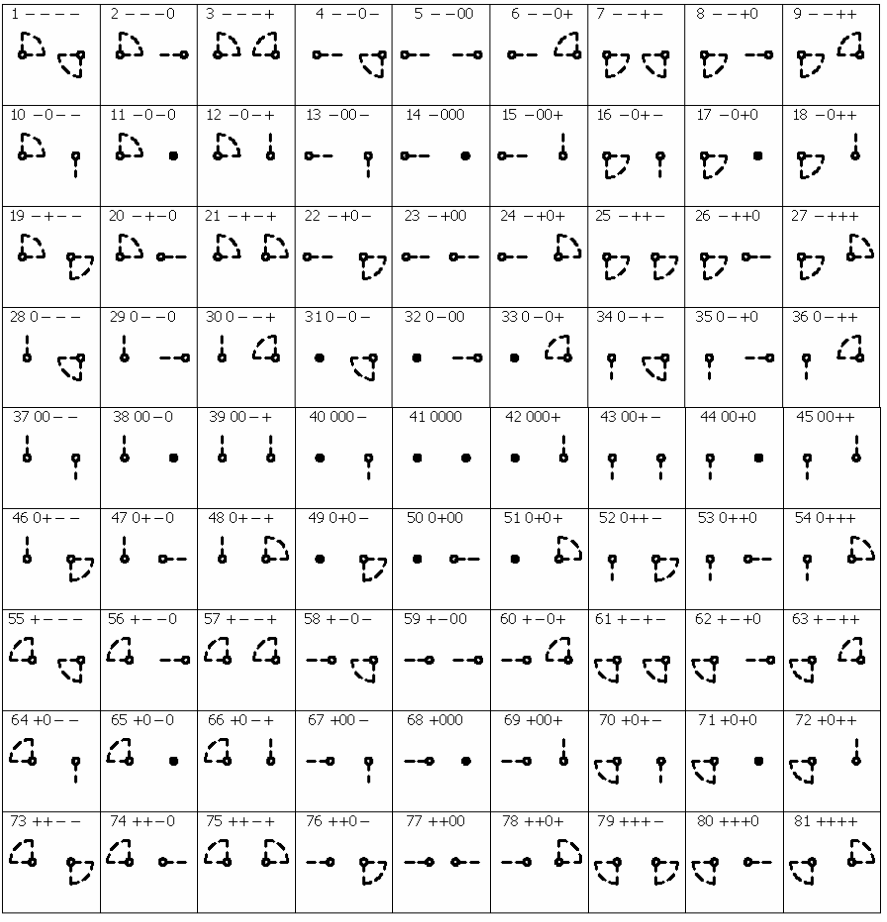


Fig. 2. QTC_S relations

$]ac$: the open ray starting at a and through c

$[ac$: the closed ray starting at a and through c

C1. Orientation of $[ab]$ wrt c (distance constraint)

−: $[ab]$ is directed towards $c \Leftrightarrow (b\alpha ac) \in]ac$ (1)

+: $[ab]$ is directed away from $c \Leftrightarrow (b\alpha ac) \notin [ac$ (2)

0: $[ab]$ is orthogonal wrt $RL_{ac} \Leftrightarrow (b\alpha ac) = a$ (3)

C2. Orientation of $[cd]$ wrt a (distance constraint)

−: $[cd]$ is directed towards $a \Leftrightarrow (d\alpha ca) \in]ca$ (4)

+: $[cd]$ is directed away from $a \Leftrightarrow (d\alpha ca) \notin [ca$ (5)

0: $[cd]$ is orthogonal wrt $RL_{ca} \Leftrightarrow (d\alpha ca) = c$ (6)

C3. Orientation of $[ab]$ wrt RL_{ac} (side constraint)

−: b is on the left side of RL_{ac} (7)

+: b is on the right side of RL_{ac} (8)

0: b is on RL_{ac} (9)

C4. Orientation of [*cd*] wrt RL_{ca} (side constraint)

−: *d* is on the left side of RL_{ca} (10)

+: *d* is on the right side of RL_{ca} (11)

0: *d* is on RL_{ca} (12)

We can represent a vector pair by a label consisting of four characters, each one giving a value for the conditions above; $(+ + + -)_S$ for the vector pair in Fig. 1. All 81 (3^4) QTC_S-relations are represented in Fig. 2 (see end of paper).[1]

Now, let us present an open oriented polyline P_1 (Fig. 3), composed of 4 straight edges $\{e_1,e_2,e_3,e_4\}$ and 5 vertices $\{v_1,v_2,v_3,v_4,v_5\}$, with $e_1 = [v_1v_2]$, $e_2 = [v_2v_3]$, $e_3 = [v_3v_4]$, and $e_4 = [v_4v_5]$. Each possible pair of edges can be represented by a QTC_S-label forming a so-called *Shape Matrix* (M_s) (Table 1a). Based on the idea of continuity, all entries of the diagonal of every M_s will be $(0\ 0\ 0\ 0)_S$. In addition, every entry from the upper part of the matrix has its inverse in the lower part of the matrix; with the inverse of the entry $(abcd)_S$ is $(badc)_S$. Therefore M_s can be reduced without information loss to a reduced M_S (see Table 1b) containing $(n^2\text{-}n)/2$ entries, *n* being the number of edges *e*.

Typically, such a boundary-based approach suffers two major problems. Firstly, when the vertices are localized strongly differently, the M_s may be exactly the same (e.g. both polylines in Fig. 4 generate the same M_s). Secondly, when the polylines count a different number of edges (see Fig. 5), the M_s have a different degree and therefore cannot be compared. To overcome these problems, we introduce the Generalization Concept, which is able to handle curved lines as well.

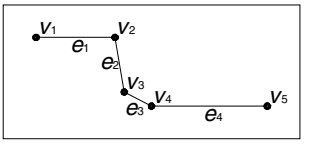


Fig. 3. Polyline P_1

Table 1. Similarity Matrix (M_S)

(a)

	e_1	e_2	e_3	e_4
e_1	0000	−+0−	−+−+	−+−+
e_2	+−−0	0000	−+0+	−+++
e_3	+−+−	+−+0	0000	−+0+
e_4	+−+−	+−++	+−+0	0000

(b)

	e_2	e_3	e_4
e_1	−+0−	−+−+	−+−+
e_2		−+0+	−+++
e_3			−+0+

[1] The left dot represents vertex *a*, the right dot vertex *b*. A dot is filled if the 'limit case' is possible (i.e.: the vector is reduced to a point) and open if the limit case is impossible. The quarter parts of circles stand for an open polygon, for which each line segment drawn from the object point to the curved side of the quarter part stands for a possibility. Note that the polygons are open, i.e. the orientation of [*ab*] in relation $(+ + + -)_S$ in Fig. 2 can be from *a* to every point on the curved part of the quarter part excluding the horizontal and the vertical line segment.

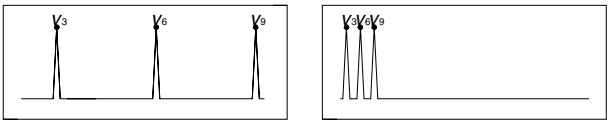


Fig. 4. First problem related to the boundary-based approach

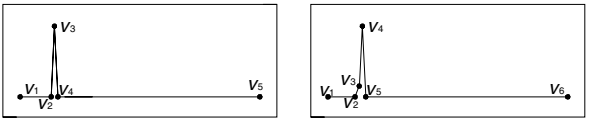


Fig. 5. Second problem related to the boundary-based approach

2.2 The Generalization Concept

Due to the Generalization Concept, QTC_S will not be based on break points (such as vertices of the polyline, extremes of curvature, and changes in curvature), but is based on so-called *division points*, the number of division points depending of the level *L*. To the best of our knowledge, this approach is innovative in the research area of shape retrieval. The algorithm is explained by use of Fig. 6a.

Assume: polyline *P* given as $\{v_1, v_2, v_3, v_4, v_5\}$ or $\{e_1, e_2, e_3, e_4\}$
$|P|$: metric length of the polyline *P* (as defined by the line integral over *P*)
$P.(v_1 \in 5)$: vertex at location 5 distance units from v_1, measured along *P*
P.d: division point of *P*
$P.d^L_n$: n^{th} *P.d* of level *L* (of the 2^L+1 division points at that level)
$P.g^L_n$: n^{th} edge of the generalization of polyline at level *L* of *P*

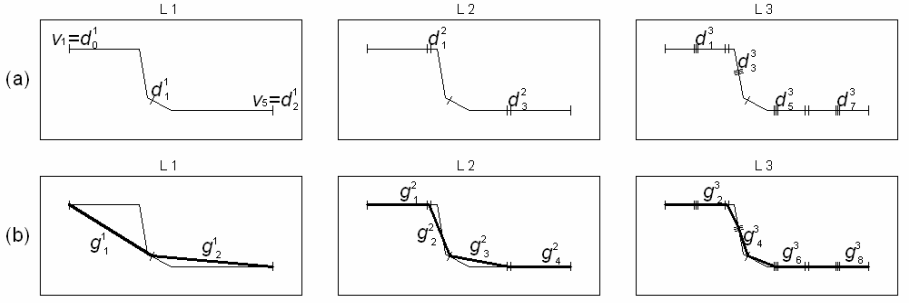


Fig. 6. Different levels of division (a) division points, (b) generalized polylines

Then: $P.d^1_0 = P.(v_1 \in 0*|P|/2)$
$P.d^1_1 = P.(v_1 \in 1*|P|/2)$ = middle of the path between v_1 and v_5 along *P*
$P.d^1_2 = P.(v_1 \in 2*|P|/2)$
$P.g^1_1 = \{P.v_1, P.d^1_1\} = \{P.d^1_0, P.d^1_1\}$
$P.g^1_2 = \{P.d^1_1, P.v_5\} = \{P.d^1_1, P.d^1_2\}$
$P.d^2_0 = P.(v_1 \in 0*|P|/4) = P.d^1_0$

$P.d^2_1 = P.(v_1 \in 1*|P|/4)$
$P.d^2_2 = P.(v_1 \in 2*|P|/4) = P.d^1_1$
$P.d^2_3 = P.(v_1 \in 3*|P|/4)$
$P.d^2_4 = P.(v_1 \in 4*|P|/4) = P.d^1_2$
$P.g^2_1 = \{P.v_1, P.d^2_1\} = \{P.d^2_0, P.d^2_1\}$
$P.g^2_2 = \{P.d^2_1, P.d^2_2\} = \{P.d^2_1, P.d^2_2\}$
$P.g^2_3 = \{P.d^2_2, P.d^2_3\} = \{P.d^2_2, P.d^2_3\}$
$P.g^2_4 = \{P.d^2_3, P.v_5\} = \{P.d^2_3, P.d^2_4\}$

In general: If L levels, then

Number of division points = 2^L+1
Number of edges = 2^L
$P.d^L_r = P.(v_1 \in r*|P|/2^L)$
$P.g^L_s = \{P.d^L_{s-1}, P.d^L_s\}$

Table 2. Similarity Matrices of different levels

L1

	g_1	g_2
g_1		-+0+
g_2		

L2

	g_1	g_2	g_3	g_4
g_1		-+0-	-+-+	-+-+
g_2			-+0+	-+++
g_3				-+0+
g_4				

L3

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
g_1		-+00	-+0-	-+--	-+-+	-+-+	-+-+	-+-+
g_2			-+0-	-+--	-+-+	-+-+	-+-+	-+-+
g_3				-+0-	-+-+	-+++	-+++	-+++
g_4					-+0+	-+++	-+++	-+++
g_5						-+0+	-+++	-+++
g_6							-+00	-+00
g_7								-+00
g_8								

Actually, each level leads to a generalization of the original polyline, increasing the complexity exponentially with increase in level. Fig. 6b and Table 2 respectively present the construction of the generalized polyline and the M_s for L1, L2, and L3. Note that, because there is a subdivision in two for every edge at every level, different similarity matrices can be compared with each other extensively:

- M_s representing the same polyline at different levels can be compared (an entry at a certain level L stands for four entries at level L+1).
- 'Analogous divisions' (i.e., analogous locations) on different polylines can be compared with each other. M_s representing different polylines at the same level can be compared, because the entries represent fixed localization on the polyline.
- By combining the two previous statements, M_s for different polylines and of different levels can be compared with each other.

Note also that, because the distance along a curved line can be measured, the algorithm can be used for polylines only containing straight edges as well as for polylines also containing curved edges.

3 Shape Similarity

It is well-known (e.g. [7] and [21-25]) that it is difficult to describe (spatial) similarity by formal logical theories, since similarity is an intuitive judgment. We therefore will

not go in detail on the mathematical aspects of similarity, but present a shape similarity measure in accordance with our visual perception, being translation, rotation, and scaling invariant. The similarity of two shapes can be assessed as the number of different entries in the M_s. We present a similarity function that maps pairs of entities onto a unique degree of similarity between 0 and 1 (with 0 corresponding to the maximum dissimilarity and 1 corresponding to the maximum similarity) [26]. In order to describe this shape similarity, we have compared four different cliffs (see Fig. 7). We used the complete-link algorithm, a well-known hierarchical agglomerative clustering algorithm. Here, the distance between two clusters is the maximum of all pairwise distances between patterns in the two clusters. One can yield a dendrogram representing the nested grouping of patterns and similarity levels at which groupings change [27]. This has been done in Fig. 8 for the four cliffs (at level 5). Fig. 8 clearly shows that Cliff 1 and Cliff 2 are very similar, and Cliff 3 and Cliff 4 as well.

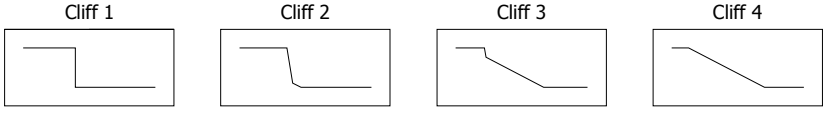


Fig. 7. Four cliffs

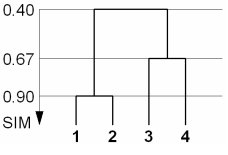


Fig. 8. Similarity between four cliffs

4 Comparison with Closely Related Calculi

Approaches, resembling mostly to our approach have been worked out in [7,13,14]. In [7], the side constraints are handled, but not the distance constraints. However, as mentioned in [28], the concept of oriented triangles for the description of arrangement of points in a two dimensional plane (as has been done in [7]) needs to be extended with the concept of qualitative angles (i.e., acute and obtuse angles), in order to get all interesting inferences. In [13] and [14], only subsequent edges are labeled having as a result that no direct qualitative relations between non-subsequent edges are known[2], a property which is often of huge importance for the (qualitative) characterization of shapes. Consider, for example,

[2] Note that, unlike the familiar arithmetic operators of the real numbers, the qualitative arithmetic operators might not give a unique value [29]. For example, if we know that the speed of a car is lower than the speed of a train, and that the speed of a bicycle is lower than the speed of the train, we cannot say whether the speed of the car is higher than the speed of the bicycle (resulting in +) or the other way round (resulting in −), or whether both objects have the same speed (resulting in 0).

the skyline of a part of a city presented in Fig. 9. If one takes, such as in QTC_S, all unique vector pairs, one detects, for example, the following things:

- e_1 and e_{13} are collinear;
- $v_6 \in v_1v_2$;
- e_1 is perpendicular to e_6[3].

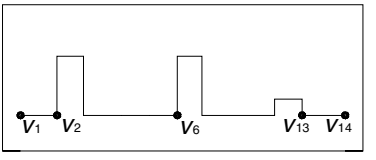


Fig. 9. Skyline of a part of a city

In fact, the remarks are much more general than only for perpendicularity and parallelism. Consider the simple, but illustrative example in Fig. 10. If one only takes the QTC_S-labels of the subsequent edges, then there is no difference between the first and the second polyline. If one uses the technique as presented in this work, then there is a clear difference between both polylines (e.g. relation in Polyline 1 between e_1 and $e_4 = (----)_S$ and relation in Polyline 2 between e_1 and $e_4 = (-+--)_S$).

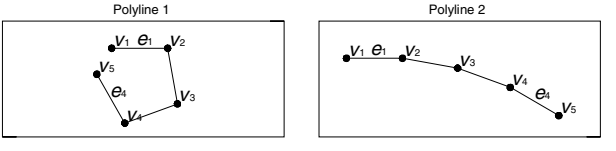


Fig. 10. Two polylines being similar when only considering subsequent edges

In addition the approach presented in [13] cannot handle curved lines, and different locations of different polylines cannot be confronted with each other. Also worth mentioning is that their primitive directly takes into account length, where the one presented in this work handles length indirectly.

5 Conclusions and Directions for Further Research

An important question is whether QTC is able to handle break points. When working with real-life coordinates having a certain precision, the number of iterations that needs to be handled can be calculated. A much faster way for handling break points is by using a so-called snapping technique. If the distance between $P.d^L_r$ and the nearest vertex of the original polyline is smaller than a certain threshold, one could force $P.d^L_r$ to snap to this vertex. This way, break points can be detected quit fast and still have a 'good' localization on the polyline, making comparisons between different polylines

[3] We do need to say that perpendicularity of edges is only detected for specific cases such as where the beginning of the second edge is an element of the line through the first edge.

straightforward. There are different sorts of snapping one might use: snap division point to vertex if distance between division point and vertex is smaller athen *x%* of PLength, or if distance between division point and vertex is smaller than *x%* or the current edge.

Another important question is whether QTC can handle closed polylines, so-called polygons, and how these will be handled? There are two options. Firstly, a polygon can be oriented and therefore can be handled as a polyline, with $v_1 = v_n$. Secondly, a polygon can be non-oriented and therefore 'every' orientation should be handled. Now, what can one understand with 'every'? One could say that there are infinite possibilities to start and end measuring a polyline. However, one could use some more or less standardized methods: one could obtain to choose for every break point as begin point. One could also obtain for a regularly rotation of *n* degrees or distance along the polygon *n*.

We have stated that the reduced M_s contains $(n^2-n)/2$ labels. Because the amount of data that needs to be handled in QTC_S-matrices will increase exponentially, the important question whether this is the maximum data reduction without information loss needs to be asked. The answer may be achieved by selecting a minimal subgraph from which one can derive the complete and original graph without losing information about the relations between the edges of the polyline. Analogous pre-processing techniques for data-reduction have been worked out for major qualitative calculi. In [31] such a pre-processing technique is presented for the calculus for reasoning about topological relations [32,33], and the data-reduction for the interval calculus for reasoning about time [34] has been worked out in [35].

Besides static forms, one could also present changes by QTC_S. Besides going from a shape to an M_s, one could also go from an M_s to a sort of shape. Both will be handled in future work. Additionally, in further research our approach has to be justified by cognitive experiments.

Acknowledgements

This research is funded by the Research Foundation - Flanders, Research Project G.0344.05.

References

1. Bookstein, F.L., 1986, Size and shape spaces for landmark data in two dimensions, *Statistical Science*, 1, 181-242.
2. Mokhtarian, F. and Mackworth, A.K., 1992, A theory of multiscale, curvature-based shape representation for planar curves, *TPAMI*, 14, 789–805.
3. Dryden, I. and Mardia, K.V., 1998, *Statistical Shape Analysis*, Wiley , 376 pp.
4. Kent, J.T. and Mardia, K.V., 2001, Shape, procrustes tangent projections and bilateral symmetry, *Biometrika*, 88, 469-485.
5. Gero, J.S., 1999, Representation and reasoning about shapes: cognitive and computational studies in visual reasoning in design, *COSIT*, 315-330.
6. Meathrel, R.C., 2001, *A General Theory of Boundary-Based Qualitative Representation of 2D Shape*, PhD Thesis, UK, University of Exeter, 239 pp.
7. Schlieder, C., 1996, Qualitative shape representation, *Geographic Objects with Indeterminate Boundaries*, Taylor & Francis, 123-140.
8. Leyton, M., 1988, A process-grammar for shape, *Artificial Intelligence*, 34,213-247.
9. Jungert, E., 1993, Symbolic spatial reasoning on object shapes for qualitative matching, *COSIT*, 444-462.

10. Latecki, L.J. and Lakämper R., 2000, Shape similarity measure based on correspondence of visual parts, *TPAMI*, 22(10), 1185-1190.
11. Sebastian, T. and Kimia, B., 2001, Curves vs skeletons in object recognition, Conf. on Image Processing, 22-25.
12. Zhang, D.S. and Lu., L., 2001, A comparative study on shape retrieval using fourier descriptors with different shape signatures, *ICIMADE*, 1-9.
13. Gottfried, B., 2003, Tripartite line tracks qualitative curve information, *COSIT*, 101-117.
14. Kulik, L. and Egenhofer M., 2003, Linearized terrain: languages for silhouette representations, *COSIT*, 118-135.
15. Van de Weghe, N., 2004, *Representing and Reasoning about Moving Objects: A Qualitative Approach*, PhD Thesis, Belgium, Ghent University, 268 pp.
16. Freksa, C., 1992, Using orientation information for qualitative spatial reasoning, COSIT, 162-178.
17. Zimmermann, K. and Freksa, C., 1996, Qualitative spatial reasoning using orientation, distance, and path knowledge, *Applied Intelligence*, 6(1), 49-58.
18. Weld, D.S. and de Kleer, J., 1990, *Readings in Qualitative Reasoning about Physical Systems*, Morgan Kaufmann, 720 pp.
19. Cohn, A.G., 1996, Calculi for qualitative spatial reasoning, *AISC*, 124-143.
20. Clementini, E., Di Felice, P. and Hernandez, D., 1997, Qualitative representation of positional information, *Artificial Intelligence*, 95(2), 317-356.
21. Winter, S., 2000, Location similarity in regions, *Journal of Photogrammetry and Remote Sensing*, 55(3), 189-200.
22. Bruns, T. and Egenhofer, M., 1996, Similarity of spatial scenes, *SDH*, 4A.31-42.
23. Nedas, K. and Egenhofer, M., 2003, Spatial similarity queries with logical operators, *SSTD*, 430-448.
24. Tversky, A., 1977, Features of similarity, *Psychological Review*, 84(4), 327-352.
25. Goyal, R.K., 2000, *Similarity Assessment for Cardinal Directions between Extended Spatial Objects*, PhD Thesis, USA, University of Maine, 167 pp.
26. Claramunt, C. and Thériault, M., 2004, Fuzzy semantics for direction relations between composite regions, *Information Sciences*, 160(1-4), 73-90.
27. Jain, A.K., Murty, M.N. and Flynn, P.J., 1999, Data clustering: a review, *Computing Surveys*, 31(3), 264-323.
28. Nabil, M., Ngu, A.H.H. and Shepherd, J., 2001, Modeling and retrieval of moving objects, *Multimedia Tools and Applications*, 13(1), 35-71.
29. Latecki, L. and Röhrig, R., 1993, Orientation and qualitative angle for spatial reasoning, *IJCAI*, 1544-1549.
30. Iwasaki, Y., 1997, Real-world applications of qualitative reasoning: introduction to the special issue, *Intelligent Systems*, 12(3), 16-21.
31. Rodríguez, A., Egenhofer, M. and Blaser A., 2003, Query pre-processing of toplogical constraints: comparing a composition-based with a neighborhood-based approach., *SSTD*, 362-379.
32. Egenhofer, M., 1991, Reasoning about binary topological relations, *SSD*, 143-160.
33. Randell, D.A., Cui, Z. and Cohn., A.G., 1992, A spatial logic based on regions and connection, *KR*, 165-176.
34. Allen, J.F., 1983, Maintaining knowledge about temporal intervals, *Comm. of the ACM*, 26(11), 832-843.
35. Rodríguez, A., Van de Weghe, N. and De Maeyer, Ph., 2004, Simplifying sets of events by selecting temporal relations, *GIScience*, 269-284.

Range and Nearest Neighbor Query Processing for Mobile Clients

KwangJin Park[1], MoonBae Song[1], Ki-Sik Kong[1], Chong-Sun Hwang[1], Kwang-Sik Chung[2], and SoonYoung Jung[3]

[1] Dept. of Computer Science and Engineering, Korea University, 5-1, Anam-dong, Seongbuk-Ku, Seoul 136-701, Korea
{kjpark, mbsong, kskong, hwang}@disys.korea.ac.kr
[2] Dept. of Computer Science and Engineering, Korea National Open University, kchung0825@knou.ac.kr
[3] Dept. of Computer Science Education, Korea University, jsy@comedu.korea.ac.kr

Abstract. Indexing techniques have been developed for wireless data broadcast environments, in order to conserve the scarce power resources of the mobile clients. However, the use of interleaved index segments in a broadcast cycle increases the average access latency for the clients. In this paper, we present the broadcast-based spatial query processing algorithm for LBS. In this algorithm, we simply sort the data objects based on their locations, then the server broadcasts them sequentially to the mobile clients. The experimental results show that the proposed BBS scheme significantly reduces the *Access latency*.

1 Introduction

The term location-based service(LBS) is a recent concept that denotes applications integrating geographic location(i.e., spatial coordinates) with the general notion of services. Examples of such applications include emergency services, car navigation systems or tourist tour planning. The field of LBS, which emerged a few years ago, presents many challenges in terms of research and industrial concerns.

In broadcast-based services, any number of clients can monitor the broadcast channel and retrieve the data as it arrives on the broadcast channel. Thus, a wireless broadcast system capable of answering LBS queries is considered a promising solution because it can serve a virtually unlimited number of users within its coverage. Furthermore, if the data is properly organized to cater to the needs of the clients, such a scheme makes effective use of the low wireless bandwidth and is ideal for achieving maximal scalability. Two key requirements for data access in wireless environments are the conservation of power and the minimization of the client waiting time. In broadcast-based systems, the mobile clients must wait until the server broadcasts the desired information. Therefore, the client waiting time is determined by the overall length of the broadcast data. In the broadcast-based model, the broadcasting of data together with an

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1097–1106, 2005.
© Springer-Verlag Berlin Heidelberg 2005

index structure is an effective way of disseminating data in a wireless mobile environment[1]. Using an index can help the client to reduce the amount of time spent listening to the broadcast channel. However, the average time which elapses between the request for the data and its receipt may be increased as a result of these additional messages. Air indexing techniques can be evaluated in terms of the following two factors. First, *Access latency* is the average time elapsed from the moment a client issues a query to the moment when the required data item is received by the client. Second, *Tuning Time* is the amount of time spent by a client listening to the channel. The *Access Latency* consists of two separate components, First, *Probe Wait* is the average duration for getting to the next index segment. second, *Bcast Wait* is the average duration from the moment the index segment is encountered to the moment when the required data item is downloaded. The *Access Latency* is the sum of the *Probe Wait* and *Bcast Wait*, and these two factors work against each other [3].

The remainder of the paper is organized as follows: Section 2 provides background information on the broadcast and air index model . Section 3 describes the proposed schemes. A performance evaluation is presented in section 4. Finally, section 5 concludes this paper.

2 Related Work

There are several challenges to be met in the development of Location Aware Query Processing [5], such as the constraints associated with the mobile environment and the difficulty of taking the user's movement into account.

Basic Index Structures. In [6] [7], a linear index structure is proposed based on the Hilbert curve, in order to enable the linear broadcasting of objects in a multi-dimensional space. The Hilbert curve needs to allocate a sufficient number of bits to represent the index values, in order to guarantee that each of the points in the original space has a distinct value. If ki is the number of bits used for a coordinate in the i-th dimension of the targeted m-dimensional space and n is the number of bits assigned to represent the coordinates, then a total of $\sum_{i=1}^{m} ki$ bits need to be allocated to represent the coordinates and the expected time for the conversion is $O(n^2)$. Besides, with this scheme, in order to identify the sequence of the broadcast data items, the clients have to wait until the index segment is arrives, even if the desired data is just in front of them. Thus, this method has the worst possible latency, because the clients have to wait until the beginning of the next index segment.

The R-tree serves as the basis of many later spatial indexing structures. All of these R-tree-based indexes share the basic assumption that spatial objects are approximated by their bounding rectangles before being inserted into the indexes. R-tree-based methods are better supported by random access storages, such as memory and disk, but not the wireless channels. Information is broadcasted based on a pre-defined and it is only available at the moment when it is broadcast. Consequently, backtracking may incur significant access latency. In

[1], the authors present a new index structure, called D-tree. Different from the existing approaches, the D-tree neither decomposes nor approximates data regions; rather, it indexes them directly based on the divisions between the regions in order to eliminate backtracking problem. However, this method has the worst latency because clients have to wait until the index segment is arrives, even if the desired data is just in front of them.

Index on Air. Data broadcasting in a wireless network constitutes an attractive approach in the mobile data environment. However, the wireless broadcast environment is affected by the narrow network bandwidth and the battery power restrictions of the mobile clients. The broadcasting of spatial data together with an index structure is an effective way of disseminating data in a wireless mobile environment. This method allows mobile clients requesting data to tune into a continuous broadcast channel only when spatial data of interest and relevance is available on the channel, thus minimizing their power consumption. Broadcasting methods that properly interleave index information and data on the broadcast channel can significantly improve not only energy efficiency, but also *Access Latency*.

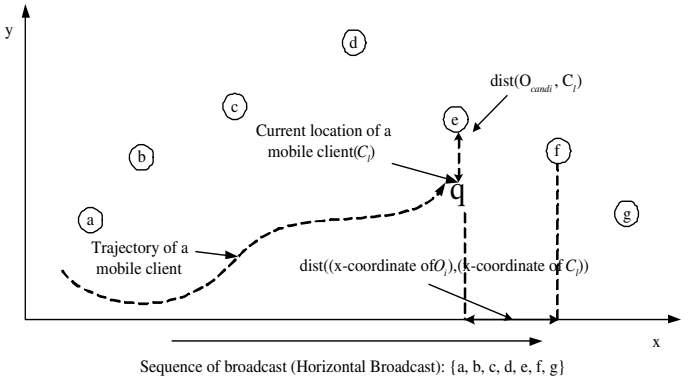
Throughout this paper, we assume that the data objects are in 2-dimensional space and are static, such as restaurants, hospitals, and hotels. The mobile clients can identify their locations using systems such as the Global Positioning System(GPS). We also assume the server periodically broadcasts data items and the client wakes up(randomly) and tune the broadcast channel, in order to process the spatial query.

3 Proposed Algorithms

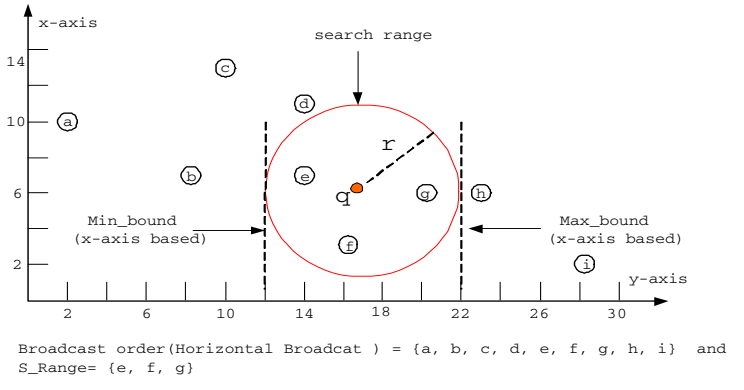
In wireless data broadcast, data items are sequentially delivered on the broadcast channel. Thus, organizing data in a way such that the clients can efficiently retrieve data is critical. In this paper, we focus on range and nearest neighbor queries.

3.1 Data Organizing

In this section, we propose a technique designed to reduce the *Access Latency*, called BBS(Broadcast-Based Sequential data delivery technique). In the BBS method, the sever periodically broadcasts the IDs and the coordinates of the data objects, along with an index segment, to the clients, and these broadcasted data objects are sorted sequentially according to the location of the data objects before being sent to the clients. In this method, since the data objects broadcasted by the server are sequentially ordered based on their location, it is not necessary for the client to wait for an index segment, if it has already identified the desired data object before the associated index segment has arrived. In this method, the structure of the broadcast affects the distribution of the data object. For example, as shown in Fig 1(a), if the data objects are horizontally distributed, the server broadcasts the data objects sequentially, from the leftmost data object to the rightmost data object. A simple sequential broadcast



(a)



(b)

Fig. 1. Detections of NN data object or data objects inside the search range

can be generated by linearizing the two dimensional coordinates in two different ways: i.e. horizontal broadcasting(HB) or vertical broadcasting(VB). In HB, the server broadcasts the LDD(location dependent data) in horizontal order, that is, from the leftmost coordinate to the rightmost coordinate, as shown in Fig 1(a). On the other hand, in VB, the server broadcasts the LDD in vertical order, that is, from the bottom coordinate to the top coordinate. In this paper, we assume that the server broadcasts the data objects using HB.

3.2 *NN* Query Processing with the BBS

In order to identify the nearest object using the BBS scheme, the client has to compare the most recently delivered data object with the previous one during the tuning time. The client uses the following algorithm to identify the nearest object:

- S: server data set
- O: broadcast data object, where $O \in S$
- O_{candi}: candidate for the nearest data object
- NN: nearest neighbor data object
- O_c: current broadcast data object (initially O_c regarded as NN), where $O_c \in S$
- O_p: previous broadcast data object, where $O_p \in S$
- C_l: client's location

Algorithm 1. the client algorithm used to identify the nearest object

```
1: if (current data item is an index segment)
2:    then find NN using an index segment
3: else
4:    do {
5:     for each data object O ∈ S {
6:        if (O_c is the first broadcast data object)
7:            then O_c = O_candi
8:        else if (dist(O_c, C_l) < dist(O_p, C_l))
9:            then O_c = O_candi
10:       else O_p = O_candi
11:   } while (getting to the index segment or dist
      ((x-coordinate of O_c),(x-coordinate of C_l))< dist(O_candi, C_l))
12:   O_candi = NN
13:   return NN
```

Lemma 3.2.1: While the data objects are sequentially broadcasted in horizontal order, that is, from the leftmost coordinate to the rightmost coordinate, if $O_c = O_i$, where $O_i \in S$ and dist((x-coordinate of O_i),(x-coordinate of C_l))>dist(O_{candi},C_l), then O_i and the rest of the broadcast data objects are located outside of the NN range.

Proof: Given a query point 'q', if the O_{candi} is an object 'e' and O_c is an object 'f', as shown in the Fig. 1(a). If dist((x-coordinate of the object 'f'),(x-coordinate of 'q'))>dist('e','q'), then the objects 'f' and 'g' are located outside of the NN range and thus the client stop tuning the broadcast channel and select the object 'e' as the NN. □

3.3 Range Query Processing with the BBS

In order to identify the desired data objects using the BBS scheme, the client has to check whether or not the current broadcast data object is inside the query range during the tuning time. Let the client's current location be point q and the object's location be point p. Let point 'q' be the center of a circle from the query range and 'r' be the radius of this circle(see Fig 1(b)). From point 'q' if the point p is located inside the circle, then p is located inside the query range. The client may change the value of 'r', in order to adjust the size of the search scope. The client uses the following algorithm to identify the desired data object:

- S: server data set
- O: broadcast data object, where $O \in S$
- O_c: current broadcast data object, where $O_c \in S$
- S_Range: set of data objects inside the search range, where $S_Range \in S$
- min_bound: minimum boundary of the x-coordinate(when the server broadcasts data using HB) of the search range from the query point 'q'.
- max_bound: maximum boundary of the x-coordinate(when the server broadcast data using HB)of the search range from the query point 'q'

Algorithm 2. the client algorithm used to identify the object inside the search range

```
1:  if(current data object is an index segment)
2:      then find objects inside of the search range using an index segment
3:  else
4:      for each data object O ∈ S {
5:          if(x-coordinate of O_c is smaller than min_bound)
6:              then O_c=O_outside
7:          else if(min_bound ≤ x-coordinate of O_c ≤ max_bound)//when broadcast using HB
8:              then compare dist(q, O_c) and dist(q, r)
9:                  if dist(q, O_c) ≤ dist(q, r)
10:                     then O_c ∈ S_Range
11:                 else O_c=O_outside
12:         else
13:             stop tune and return the S_Range
14:     }
```

Lemma 3.3.1: If x-coordinate of $O_c <$ min_bound, then O_c is located outside of the search range.

Proof: Given a query point 'q' and the search range radius 'r', let O_c is an object 'b' in Fig 1(b). Since the x-coordinate of O_c is outside of the circle, then O_c is located outside of the search range. □

Lemma 3.3.2: When the data objects are sequentially broadcasted in horizontal order, that is, from the leftmost coordinate to the rightmost coordinate, if the x-coordinate of $O_c > maxbound$, then O_c and the rest of the broadcast data objects in the cycle are located outside of the search range.

Proof: Given a query point 'q' and the search range radius 'r', let O_c be an object 'h'(see Fig 1(b). Since O_c and the rest of the broadcast objects' x-coordinates are greater than max_bound, object 'h' and 'i' are located outside of the search range. Thus, the client no longer needs to tune into the broadcast channel, and return $S_Range = \{e, f, g\}$. □

Lemma 3.3.3: The results of the **Lemma 3.3.1** and the **Lemma 3.3.2** lead us to the conclusion that if O_c satisfies the following steps, then $O_c \in S_Range$.

Step 1: The x-coordinate of O_c satisfies: min_bound$\leq$ x-coordinate of $O_c \leq$ max_bound
Step 2: $dist(q, O_c) \leq dist(q, r)$.

The following shows comparison of the *Probe Wait* and the *Bcast Wait* between BBS and previous index method [3]. Let m denotes the number of times broadcast indices:

- *Probe Wait* of previous index method: $\frac{1}{2} * (index + \frac{data}{m})$
- *Probe Wait* for BBS method: $None$
- *Bcast Wait* of previous index method: $\frac{1}{2} * ((m * index) + data) + C$
- *Bcast Wait* of BBS method: $\frac{1}{2} * ((m * index) + data) + C$

Since the *Access Latency* is the sum of the *Probe Wait* and the *Bcast Wait*, average *Access Latency* for:

- **Previous index method** is $\frac{1}{2} * ((m + 1) * index + (\frac{1}{m} + 1) * data) + C$
- **BBS** is: $\frac{1}{2} * ((m * index) + data) + C$.

The following shows the average tuning time of BBS. Let m denotes the number of times broadcast indices, k' denotes the number of levels in the index tree for BBS and P denotes the probability:

P(containing the desired data object among the index) is $\frac{1}{m}$, and then, P(obtaining the desired data object) is $\frac{1}{2m}$. Thus, average tuning time is P(obtaining data object without an index)* cost of reading data objects + P(failure obtaining the desired data object after read the index)* cost of obtain the desire data object after read the index, and thus,

$$\begin{aligned} f(m) &= \frac{1}{2m}\left(\frac{Data}{m} * \frac{1}{2}\right) + \frac{2m-1}{2m}\left(\frac{Data}{m} * \frac{1}{2} + k' + 1\right) \\ &= \frac{Data - k' - 1}{2} m^{-1} + k' + 1 \end{aligned} \tag{1}$$

Now, we present a formula to compute the optimal m so as to minimize the latency and tuning time for the $(1, m)$ indexing [3]. For finding the minimal latency and tuning time, we differentiate the above formula with respect to m, equate it to zero and solve for m; m^* denotes the optimum m.

$$\begin{aligned} f(m) &= Access\ Latency + Tuning\ Time \\ &= \frac{index * m}{2} + \frac{Data - k' - 1}{2m} + \frac{Data + k' + 2C}{2} \\ &= f'(m) = index - \frac{Data - k' - 1}{1} m^{-2} = 0 \end{aligned} \tag{2}$$

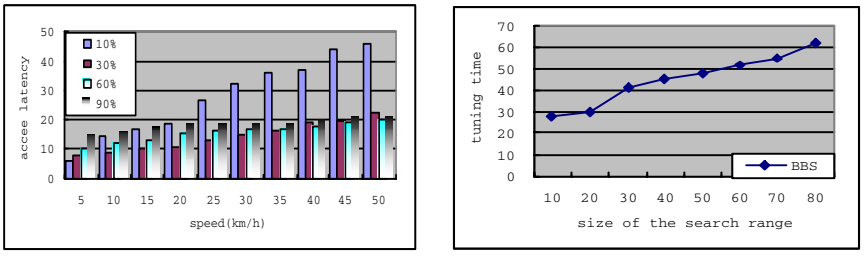
$$m^* = \sqrt{\frac{Data - k' - 1}{Index}} \tag{3}$$

4 Performance Evaluation

We use a system model similar to that in [11]. The whole geometric service area is divided into groups of MSSs (Mobile Supporting Stations). The data objects of hospitals in the Southern California area, which is extracted from the data set at [14] are used for performance evaluation.

4.1 Access Latency and Tuning Time

First, we evaluate the *Access Latency*. We vary the client's speed from 5 to 50. When the client's speed is the lowest, broadcast size of 10% (of the coverage area) is the best. However, as the client's speed increases, its performance is degraded in comparison with that of the other clients, since for most values of the client's speed, the client is outside of the service coverage area, as shown in Fig. 2(a). Then, we evaluate the *Tuning Time*. We study the effect of the size of the search range. We vary the size of the search range from 10km to 80km. Fig. 2(b) shows the *Tuning Time* of the BBS as the size of the search range increases. As shown in the figure, client's *Tuning Time* increases as the size of the search range increase.



(a) (b)

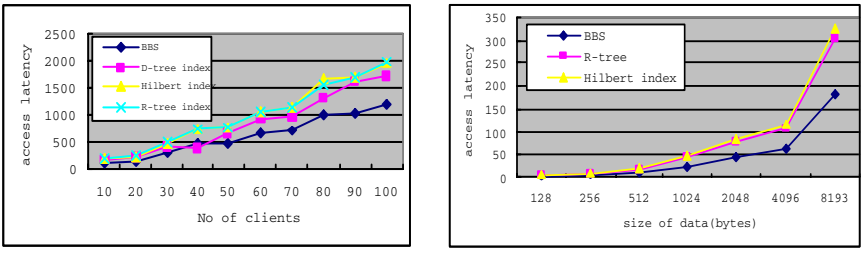
Fig. 2. *Access latency and Tuning Time*

4.2 Comparison of the Performance of the BBS Scheme with the R-Tree, the D-Tree Index and the Hilbert-Curve Index

In this section, we compare the BBS scheme with the R-Tree, the D-Tree index schemes and Hilbert-Curve index. We assume the access latency of the R-tree introduces the access latency about 1.23 and Hilbert curve index incurs the latency about 1.30 same as [7].

First, we vary the number of the clients from 10 to 100. In this experiment, the server periodically broadcasts 1000 data objects to the clients. Since the clients do not need to wait and tune into the broadcast channel to receive an index segment, if they have already identified the desired objects, the BBS scheme

shows increasingly lower latency compared to the R-Tree ,the D-tree index and the Hilbert-curve index scheme. Fig 3(a) shows the result as the number of client increases. As shown in the figure, as the number of the client increases, BBS outperforms the other schemes, since it is not necessary for the client to wait for an index segment, if it has already identified the desired data object before the associated index segment has arrived. Fig 3(b) shows the result of the *Access Latency* as the size of data increases from 128 to 8193 bytes. In the same way of the BBS, the Hilbert-curve index scheme also broadcasts data items based on their locations. The Hilbert-curve maps points from a multi-dimensional space to a one-dimensional space according to the localities of the data objects. However, with this scheme, in order to identify the sequence of the broadcast data items, the clients have to wait until the index segment is arrives, even if the desired data is just in front of them. Thus, this method has the worst possible latency, because the clients have to wait until the beginning of the next index segment.



(a) (b)

Fig. 3. Compare the Performance of BBS Scheme with the R-Tree, D-Tree and the Hilbert-curve index scheme

5 Conclusion

In this paper, we have studied the different broadcasting schemes which can be used for NN or range query processing. In these methods, we do not modify the previous index scheme per se. Rather, we simply sort the data objects based on their locations, and the server then broadcasts them sequentially to the mobile clients. The BBS method attempts to reduce the *Access Latency* for the client. With the proposed schemes, the client can perform NN or range query processing without having to tune the index segment. The experimental results show that the proposed BBS scheme significantly reduces the *Access latency* compared with the R-tree index and D-tree schemes, since the client does not always have to wait for the index segment. In this paper, we do not consider the case of moving data objects in LAMSs. Hence, we are planning to extend this study to the case of a moving object database.

References

1. J. Xu, B. Zheng, W.-C. Lee, and D. L. Lee, "D-tree: An Index Structure for Planar Point Queries in Location-Based Wireless Services," In *IEEE Trans. on Knowledge and Data Engineering (TKDE)*, 2004, 16(12): 1526-1542.
2. T. Imielinski, S. Viswanathan, and B.R.Badrinath, "Energy efficient indexing on air," In *Proc. of SIGMOD*, 1994, pp. 25-36.
3. T. Imielinski, S. Viswanathan, and B.R.Badrinath, "Data on Air: Organization and Access," *IEEE Trans. Knowledge and Data Eng*, 1997, 9(3): 353-372.
4. T. Camp, J. Boleng and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," In *Wireless Communication and Mobile Computing(WCMC)*, 2002, 2(5): 483-502.
5. D. L. Lee, J. Xu, and B. Zheng, "Data Management in Location-Dependent Information Services," *IEEE Pervasive Computing*, 2002, 1(3): 65-72.
6. B. Zheng, W.-C. Lee, D. L. Lee, "Search Continuous Nearest Neighbor on Air," In *Proc. The First International Conference on Mobile and Ubiquitous Systems: Networking and Services*, Boston, Massachusetts, 2004, pp. 22-26.
7. B. Zheng, W.-C. Lee, D. L. Lee, "Spatial Queries in Wireless Broadcast Systems," In *Wireless Networks*, 2004, 10(6): 723-736.
8. W. C. Lee and D. L. Lee, "Using signature techniques for information filtering in wireless and mobile environments," *Distributed and Parallel Databases*, 1996, 4(3): 205-227.
9. Q. L. Hu, W.-C. Lee, and D. L. Lee, "A hybrid index technique for power efficient data broadcast," *Distributed and Parallel Databases*, 2001, 9(2): 151-177.
10. J. Xu, X. Tang, and D. L. Lee, "Performance Analysis of Location-Dependent Cache Invalidation Schemes for Mobile Environments," *IEEE Trans. Knowledge and Data Eng*, 2003, 15(2): 474-488.
11. B. Zheng, J. Xu, and D. L. Lee, "Cache Invalidation and Replacement Strategies for Location-Dependent Data in Mobile Environments," *IEEE Trans. Comp.*, 2002, 51(10): 1141-1153.
12. A. Guttman, "R-trees: A dynamic index structure for spatial searching," In *Proc. of SIGMOD*, 1984, pp. 599-609.
13. D. Barbara and T. Imielinski, "Sleepers and Workaholics: Cashing Strategies in Mobile Environments," In *Proc. of SIGMOD*, 1994, pp. 1-12.
14. Spatial Datasets, `http://dias.cti.gr/ ytheod/research/datasets/spatial.` html.
15. N. Roussopoulos, S. Kelley, and F. Vincent. "Nearest neighbor queries," In *Proc. of SIGMOD*, 1995, pp. 71-79.
16. J. Xu, B. Zheng, W.-C. Lee, and D. L. Lee, "Energy Efficient Index for Querying Location-Dependent Data in Mobile Broadcast Environments," In *Proc. of ICDE*, 2003, pp. 239-250.

An Efficient Trajectory Index Structure for Moving Objects in Location-Based Services*

Jae-Woo Chang[1], Jung-Ho Um[1], and Wang-Chien Lee[2]

[1] Dept. of Computer Eng., Chonbuk National Univ.,
Chonju, Chonbuk 561-756, Korea
{jwchang, jhum}@dblab.chonbuk.ac.kr
[2] Dept. of CS&E., Pennsylvania State Univ.,
University Park, PA 16802

Abstract. Because moving objects usually moves on spatial networks, efficient trajectory index structures are required to gain good retrieval performance on their trajectories. However, there has been little research on trajectory index structure for spatial networks, like road networks. In this paper, we propose an efficient trajectory index structure for moving objects in Location-based Services (LBS). For this, we design our access scheme for efficiently dealing with the trajectories of moving objects on road networks. In addition, we provide both an insertion algorithm to store the initial information of moving object trajectories and one to store their segment information. We also provide a retrieval algorithm to find a set of moving objects whose trajectories match the segments of a query trajectory. Finally, we show that our trajectory access scheme achieves about one order of magnitude better retrieval performance than TB-tree.

1 Introduction

In general, spatial databases has been well studied in the last two decades, resulting in the development of numerous spatial data models, query processing techniques, and index structures for spatial data [Se99]. Most of the existing work considers Euclidean spaces, where the distance between two objects is determined by the ideal shortest path connecting them in the spaces. However, in practice, objects can usually move on road networks, where the network distance is determined by the length of the real shortest path connecting two objects on the network. For example, a gas station nearest to a given point in an Euclidean space may be more distant in a road network than another gas station. Therefore, the network distance is an important measure in spatial network databases (SNDB). Recently, there have been some studies on SNDB for emerging applications such as location-based service (LBS) [B02, HKT02, F03, PZM03, SJK03, SKS03]. First, Speicys et al. [SJK03] dealt with a computational data model for spatial network. Secondly, Shahabi et al. [SKS03] presented k-nearest neighbors (k-NN) query processing algorithms for SNDB. Finally, Papadias et al. [PZM03] designed a novel index structure for supporting query processing algorithms.

* This work was supported by the Korea Research Foundation Grant. (KRF-2003-013-D00094).

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1107 – 1116, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Because moving objects usually moves on spatial networks, instead of on Euclidean spaces, efficient index structures are required to gain good retrieval performance on their trajectories. However, there has been little research on trajectory access schemes for spatial networks, like road networks. In this paper, we propose an efficient trajectory index structure for moving objects in Location-based Services (LBS). For this, we design our access scheme for efficiently dealing with the trajectories of moving objects on road networks. In addition, we provide both an insertion algorithm to store the initial information of moving object trajectories and one to store their segment information. We also provide a retrieval algorithm to find a set of moving objects whose trajectories match the segments of a query trajectory.

The rest of the paper is organized as follows. In Section 2, we discuss background and motivation for our work. In Section 3, we propose a trajectory access scheme for moving objects. In Section 4, we provide the performance analysis of our access scheme. Finally, we draw our conclusion in Section 5.

2 Background and Motivation

To our knowledge, there has been little research on trajectory access schemes for spatial networks. So we overview both a major index structure for spatial networks and a predominant trajectory index structure for Euclidean spaces. First, Papadias et al. [PZM03] proposed a network storage scheme for spatial network databases (SNDB) by separating the network itself from entity datasets. They employ a disk-based network representation that preserves connectivity and location, while spatial entities are indexed by respective spatial access methods for supporting Euclidean queries and dynamic updates. Their network storage scheme consists of three components, i.e., adjacency component, poly-line component, and network R-tree. The adjacency component captures the network connectivity by which the adjacency lists of nodes being close in space according to their Hilbert values are placed in the same disk. The poly-line component stores the detailed poly-line representation of each segment in the network. The last network R-tree component indexes the poly-lines' MBRs (Minimum Bounding Rectangles) and supports queries exploiting the spatial properties of the network. In addition, Pfoser et al. [PJT00] proposed a hybrid index structure which preserves trajectories as well as allows for R-tree typical range search in Euclidean spaces, called TB-tree (Trajectory-Bundle tree). The TB-tree has fast accesses to the trajectory information of moving objects, but it has a couple of problems in SNDB. First, because moving objects move on a predefined spatial network in SNDB, the paths of moving objects are overlapped due to frequently used segments, like downtown streets. This leads to a large volume of overlap among the MBRs of internal nodes. Thus, the TB-tree no longer achieves good performance on retrieving moving object trajectories in SNDB. Secondly, when a moving object moves on the predefined spatial network for a long time, the dead space for the moving object trajectory is highly increased because the TB-tree constructs a three-dimensional MBR including time. This leads to a large volume of overlap with other objects' trajectories. Thus, the TB-tree results in poor retrieval performance on moving object trajectories in SNDB, due to its small discrimination capability.

3 Trajectory Access Scheme for Moving Objects

3.1 Architecture of Trajectory Access Scheme

Because moving objects change their locations continuously on road networks, the amount of trajectory information for a moving object is generally very large. To solve the problems of TB-tree as mentioned in section 2, we propose a new signature-based access scheme which can have fast accesses to moving object trajectories. Figure 1 shows the structure of our signature-based trajectory access scheme.

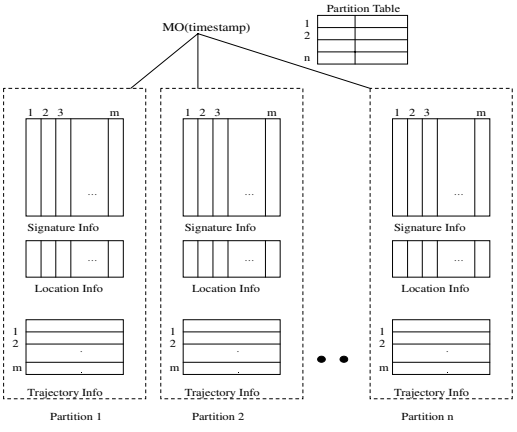


Fig. 1. Architecture of trajectory access scheme

The main idea of our trajectory access scheme is to create and maintain partitions which store the predefined number of moving object trajectories and their signatures together in the order of their start time. There are a couple of reasons for using partitions. First, because a partition includes the fixed number of moving object trajectories, it is possible to construct a bit-sliced signature file which enables good retrieval performance. Secondly, because a partition can be accessed independently to answer a trajectory-based query, it is possible to achieve better retrieval performance by searching partitions in parallel. Finally, because a partition is created and maintained depending on its start time, it is possible to efficiently retrieve the trajectories of past moving objects on a given time. Therefore, our trajectory access scheme has three advantages. First, our access scheme is not affected by the overlap of moving objects' paths and never causes the dead space problem because it is not a tree-based structure like TB-tree. Secondly, our access scheme well supports a complex query containing a partial trajectory condition since it generates signatures using a superimposed coding. Finally, our access scheme can achieve very good retrieval performance because it can be easily adapted to a parallel execution environment.

Our trajectory access scheme consists of a partition table and a set of partitions. A partition can be divided into three areas; trajectory information, location information,

and signature information. A partition table maintains a set of partitions which store trajectories for current moving objects. The partition table is resided in a main memory due to its small size. To answer a user query, we find partitions to be accessed by searching the partition table. An entry Ei for a partition i is the following.

Ei = <p_start_time, p_end_time, p_expected_time, final_entry_no>
- p_start_time, p_current_time, p_end_time : the smallest start time, the largest current and end time of all the trajectories in a partition I, respectively
- final_entry_no : the last entry number in a partition i

Trajectory information area maintains moving object trajectories which consist of a set of segments (or edges). A trajectory Ti for an object MOi is the following.

Ti =<MOiid,#_past_seg,#_future_seg,#_mismatch, {sij,eid,start,end,ts,(te or v)}>
- MOiid : ID for MOi
- #_past_seg, #_future_seg: the number of past and expected future segments
- #_mismatch: the # of mismatched segments between past and future segments
- sij: j-th segment of the trajectory for MOi, eid: edge ID for an edge covering sij
- start, end: relative start and last location of sij in the edge of eid, respectively
- ts, te: start and end time of sij in the edge of eid, respectively
- v : average speed of sij in the edge of eid in case of a future segment

The location information area contains the location of an object trajectory stored in the trajectory information area. This allows for accessing the actual object trajectories corresponding to potential matches to satisfy a query trajectory in the signature information area. The location information area also allows for filtering out irrelevant object trajectories based on the time condition of a query trajectory because it includes the start time, the current time, and the end time for a set of object trajectories. A location information, Ii, for the trajectory of an object MOi is the following.

Ii = <MOiid, Li, strat_time, current_time, end_time >
- Li : location for MOi in the trajectory information area
- start_time : time when the trajectory for MOi is first inserted
- current_time : time when the last segment of the trajectory for MOi is inserted
- end_time : time when the last expected segment for MOi will be inserted

To construct our trajectory access scheme, it is necessary to create a signature from a given object trajectory in an efficient manner. To achieve it, we make use of a superimposed coding because it is very suitable to SNDB applications where the number of segments for an object trajectory is variable [ZMR98]. In case the total number of object trajectories is N and the average number of segments per object trajectory is r, optimal values for both the size of a signature in bits (S) and the number of bits to be set per segment (k) can be calculated. Fd is a false drop probability that a trajectory signature seems to qualify, given that the corresponding object trajectory does not actually qualify. Assuming that Fd is 1/N, we can calculate the optimal values for S and k by using such formulas as ln Fd = - (ln 2)2 *S/r and k = S * ln 2/r [FC84]. In order that our signature-based access scheme should achieve good retrieval performance, we make use of a bit-sliced signature method for constructing our trajectory access scheme [ZMR98]. In the bit-sliced method, we create a fixed-length signature slice for each bit position in the original signature string. That is, we store a set of the

first bit positions of all the trajectory signatures into the first slice, a set of the second bit positions into the second slice and so on. In the bit-sliced method, when the number of segments in a query trajectory is m and the number of bits assigned to a segment is k, the number of page I/O accesses for answering the query in our signature-based access scheme is less than k*m. Our access scheme has a couple of advantages from the viewpoint of retrieval performance. First, when the number of segments in a query trajectory is small, our access scheme requires the small number of page I/O accesses due to the small number of signature slices needed for the query. Secondly, as the number of segments in a query trajectory is increased, the number of page I/Os to be accessed in the signature information area is increased, but the number of page I/O accesses in the trajectory information area is decreased. Thus, when the number of segments in a query trajectory is large, our signature-based access scheme achieves good retrieval performance eventually.

When a moving object trajectory is assumed to be hardly used for a trajectory-based query, the object trajectory should be moved to the past object trajectory structure (POTS) which is maintained in a tertiary storage. We make use of a strategy to move past object trajectories to POTS in a minimize cost by means of storing them into POTS with the unit of partition. In this case, the partition table maintains the information of all the partitions which will be answered for a trajectory-based query.

3.2 Insertion Algorithms

The algorithms for inserting moving objects trajectories can be divided into an initial trajectory insertion algorithm and a segment insertion algorithm for its trajectory. For the initial trajectory insertion, we find the last partition in the partition table and obtain an available entry (NE) in the last partition. To insert the initial trajectory with no expected trajectories, we create a new expected future segment based on an edge where an object currently moves and store it into the NE entry of the trajectory information area in the last partition. Using the expected future segment created, we store start_time(StartT), current_time(CurrentT), and end_time (ExpectedET) into the NE entry of the location information area in the last partition. Here StartT and CurrentT are both assigned to the start time of the moving object and ExpectedET is assigned to NULL. To insert the initial trajectory with expected trajectories, we insert a list of expected future segments (TrajSegList) into the NE entry of the trajectory information area in the last partition. We create a segment signature (SSn) from each of TrajSegList and generate a trajectory signature (SigTS) by using superimposing (Oring) all of the segment signatures. Using the TrajSegList, we store StartT, CurrentT, and ExpectedET into the NE entry of the location information area. ExpectedET is assigned to the expected end time of the last segment of the TrajSegList. Finally, we store the SigTS into the NE entry of the signature information area in the last partition, by using the bit-sliced manner. We store <StartT, CurrentT, ExpectedET> into the last partition entry (LP) of the partition table. Figure 2 shows the initial trajectory insertion algorithm (i.e., InsertFirst) for a moving object.

To insert the segment of a moving object trajectory, we find a partition storing its trajectory from the partition table by using the start time (ST) of the moving object. We obtain the entry (NE) storing the trajectory information of the partition, covering

```
Algorithm InsertFirst(MOid, TrajSegList)
/* TraSegList contains the information of a set of expected segments for
the trajectory of a moving object Moid */
1.  TrajSeg = the first segment of TrajSegList
2.  Generate a signature SigTS from TrajSeg
3.  StartT = CurrentT = ts of TrajSeg
4.   Obtain final_entry_no of the entry, in the partition table, for the
     last partition, LP
5.   NE = final_entry + 1 //NE=the next available entry in LP
6.   Obtain the location, Loc, of the entry NE in the trajectory info
     area for inserting object trajectory
7.    if(end field of TrajSeg=NULL){//no expected trajectory
8.      ExpectET = NULL
9.      Store <MOid,0,1,TrajSeg> into the entry NE, pointed by Loc, of
          the trajectory information area in LP}
10.  else { // expected trajectory exists
11.     #fseg = 1
12.     while (the next segment Sn of TrajSegList • NULL) {
13.         #fseg = #fseg + 1
14.         Generate a signature SSn from Sn
15.         SigTS = SigTS | SSn }
16.     Store <MOid,0,#fseg,TrajSegList> into the entry NE, pointed by
          Loc,
        of the trajectory info area in LP
17.    Compute ExpectET by using ts, start, and v of the last segment of
        TrajSegList } // end of else
18. Store SigTS, using the bit-sliced manner, into the entry NE of the
     signature information area in LP
19.  Store <MOid,Loc,StartT,CurrentT,ExpectET> into the entry NE of the
     location information area in the LP
20.  Store <StartT,CurrentT,ExpectET,NE> into the entry for LP in the
     partition table
End InsertFirst
```

Fig. 2. Initial trajectory insertion algorithm for moving objects

the object identified by MOid. To insert a segment for trajectories with no expected future ones, we store a new segment (TrajSeg) into the NE entry of the trajectory information area, being addressed by Loc. Then, we generate a trajectory signature (SigTS) from the TrajSeg and store the SigTS into the NE entry of the signature information area in the bit-sliced manner. Finally, we store <MOid, Loc, StartT, CurrentT, ExpectET> into the NE entry of the location information area. To insert a segment for trajectories with expected future ones, we can store a new segment according to three types of the discrepancy between a new segment and the expected segment of a trajectory by calling a function named find-seg(). First, in case of no segment coinciding with TrajSeg (seg_pos = 0), we perform the same procedure as the segment insertion algorithm with no expected future segments. Secondly, in case where the segment coinciding with TrajSeg is the first one (seg_pos = 1), we just store the TrajSeg into the (#_actual_seg)-th segment of the NE entry. Finally, in case where the segment coinciding with TrajSeg is not the first one (seg_pos > 1), we delete seg_pos-1 segments from the expected segments of the NE entry, store the TrajSeg into the (#_actual_seg)-th segment of the NE entry, and move all the expected segments of the NE entry forward by seg_pos-2. If the ratio of mismatched segments (#_mismatch) over all the segments of the trajectory is not greater than a threshold (τ), we store the trajectory signature (SigTS) generated from the TrajSeg into the NE entry of the signature information area in the bit-sliced manner. If the discrepancy is greater than τ, we regenerate SigTS from the trajectory information

of the NE entry and store all the bits of the SigTS into all the bit-sliced signatures in the signature information area in the partition P. Finally, we update the values of #_actual_seg, #_future_seg, and #_mismatch in the NE entry and update the CurrentT of the NE entry in the location information area as well as that of the partition P's entry in the partition table. Figure 3 shows the segment insertion algorithm (i.e., InsertSeg) for moving object trajectories.

```
Algorithm InsertSeg(MOid, TrajSeg, ST) /* TraSeg contains a segment for
the trajectory of a moving object Moid, to be stored with an object
trajectory's start time, ST*/
1.  Generate a signature SigTS from TrajSeg
2.  Locate a partition P covering ST in partition table
3.  Locate an entry E covering ST for the moving object with MOid in
    the location information area and get its location, Loc, in the
    trajectory information area
4.  Obtain #actual_seg, #future_seg, and #mismatch of the trajectory
    info entry E (i.e., TE) for the MOid in P
5.  if(#future_seg = 0) { // no expected trajectory
6.        Insert TrajSeg into the (#actual_seg+1)-th trajectory segment
          of TE
7.        Store SigTS, using the bit-sliced manner, into the entry E of
          the signature info area in P}
8.  else { // expected trajectory exists
9.        seg_pos = find_seg(TrajSeg,Loc)
10.       #actual_seg++, #future_seg = #future_seg - seg_pos
11.       case(seg_pos = 0) { // find no segment
12.           Insert TrajSeg into segment of TE and relocate the future
              traj segments backward
13.           Store SigTS, using the bit-sliced manner, into the entry E
              of the signature info area in P }
14.       case(seg_pos = 1) //find the first segment
15.           Insert TrajSeg into (#actual_seg)-th trajectory segment of
              TE for exchanging the old segment
16.       case(seg_pos > 1) {//find the (seg_pos)-th segment
17.           #mismatch = #mismatch + seg_pos - 1
18.            Insert TrajSeg into (#actual_seg)-th segment of TE and
               relocate the future traj segments forward
19.           if(#mismatch/(#future_seg+#actual_seg) > •)
              regenerate_sig(Loc,SigTS,E,P)}}// end of case// end of else
20. Update #actual_seg, #future_seg, and #mismatch of TE
21. CurrentT = te of TrajSeg
22. Store CurrentT into the current_time of the entry E of the location
    information area in the partition P
23. Store CurrentT into the p_current_time of the partition P entry in
    the partition table
End InsertSeg
```

Fig. 3. Segment insertion algorithm for moving object trajectories

3.3 Retrieval Algorithm

The retrieval algorithm for moving object trajectories finds a set of objects whose trajectories match the segments of a query trajectory. To find a set of partitions satisfying the time interval (TimeRange) represented by <lower, upper> of a given query (Q), we call a find_partition function to generate a list of partitions (partList) by performing the sequential search of the partition table and find a list of partitions (partList) to satisfy the following condition 3. Next, we generate a query signature

(QSig) from a query trajectory's segments. For each partition of the partList, we search only the signature slices corresponding to the bits set by '1' in QSig, in the signature information area. We create a list of candidates (CanList) being set to '1' by bit-oring the signature slices. For the entries corresponding to the candidates, we determine if their start_time, end_time, and current_time satisfy the following condition 3. Finally, we determine if the query trajectory matches the object trajectories corresponding to the entries. The matching means that the object trajectory's segments contain those of the query trajectory. If the matching is found, we insert the object 's ID into a result list (MoidList). Figure 4 shows the retrieval algorithm (i.e., Retrieve) for moving object trajectories.

$$\begin{aligned}&(\text{end_time} \geq \text{T.lower}) \text{ AND } (\text{start_time} \leq \text{T.upper}) \text{ if end_time} \neq \text{NULL}\\ &(\text{current_time} \geq \text{T.lower}) \text{ AND } (\text{start_time} \leq \text{T.upper}) \text{ otherwise}\end{aligned} \quad (3)$$

```
Algorithm Retrieve(QSegList, TimeRange, MOidList) /* MOidList is a set
of ids of moving objects containing a set of query segments, QsegList,
for a given range time, TimeRange */
1.  Qsig = 0, #qseg = 0, partList = ∅
2.  t1 = TimeRange.lower, t2 = TimeRange.upper
3.  for each segment QSj of QsegList {
4.       Generate a signature QSSi from Qsj
5.       QSig = QSig | QSSj, #qseg = #qseg + 1 }
     /*find partitions, partList, satisfying TimeRange by searching
     patiotion table of COTSS and B+-tree of POTSS*/
6.  find_partition(TimeRange, partList)
7.  for each partition Pn of partList {
8.       Obtain a set of candidate entries, CanList, by examining the
          bit slices of the signature info area in Pn, corresponding to
          bits set by 1 in QSig
9.       for each candidate entry Ek of CanList {
10.      Let s,e,c be start_time, end_time, current_time ofthe entry Ek
          of location information area
11.      if((s • t2) AND (e • t1 OR c • t1)){
12.       #matches = 0
13.       Obtain the first segment ESi of the entry Ek of the trajectory
          info area, TEk
14.       Obtain the first segment QSj of QsegList
15.       while(ESi • NULL and QSj • NULL) {
16.       if(match(Esi, QSj)=FALSE)
               Obtain the next segment ESi of TEk
17.       else { #matches = #matches + 1
18.       Obtain the first segment ESi of Tek }
19.       if(#matches=#qseg)MOidList=MOidList ∪ {TEk's MOid}
20.       } } } //end of while //end of if //end of for- CanList
21. } // end of for - partList
End Retrieve
```

Fig. 4. Retrieval algorithm for moving object trajectories

4 Performance Analysis

We implement our trajectory access scheme under Pentium-IV 2.0GHz CPU with 1GB main memory, running Window 2003. For our experiment, we use a road network consisting of 170,000 nodes and 220,000 edges [WMA]. For simplicity, we consider bidirectional edges; however, this does not affect our performance results. We also generate 50,000 moving objects randomly on the road network by using

Brinkhoff's algorithm [B02]. For performance analysis, we compare our access scheme with TB-tree which is well known as an efficient trajectory access scheme for Euclidean spaces [PJT00], in terms of insertion time and retrieval time for moving object trajectories. First, Table 1 shows the insertion performance to store one moving object trajectory.

Table 1. Insertion performance

	TB-tree	Our signature-based scheme
Trajectory insertion time(ms)	0.03	0.95

It is shown from the result that our access scheme provides much worse insertion performance than TB-tree. The reason is because when a segment for a moving object trajectory is inserted, our access scheme needs to acquire a signature built for the moving object trajectory, change the signature, and store it in bit-sliced manner. On the contrary, TB-tree only needs to append the segment to the existing moving object trajectory.

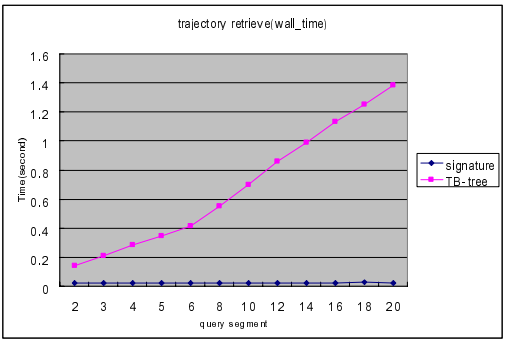


Fig. 5. Retrieval performance

Next, we measure retrieval time for answering queries whose trajectory contains 2 to 20 segments. Figure 5 shows the retrieval time of our trajectory access scheme and TB-tree. It is shown from the result that our access scheme requires about 20 ms while TB-tree needs 140ms, when the number of segments in a query is 2. Thus our access scheme nearly 700% outperforms TB-tree. Furthermore, as the number of segments in queries increases, the retrieval time is increased in TB-tree; however, our access scheme requires constant retrieval time. When the number of segments in a query is 20, it is shown that our access scheme requires about 24 ms while TB-tree needs 1380ms. Thus our access scheme achieves about two orders of magnitude better retrieval performance than TB-tree. The reason is why our trajectory access scheme creates a query signature combining all the segments in a query and searches for potentially relevant trajectories of moving objects once by using the query signature as a filter because it generates signatures based on a superimposed coding technique. On the contrary, TB-tree achieves bad performance due to a large extent of overlap in its internal nodes when the

number of segments in a query trajectory is small. TB-tree doesn't achieve good retrieval performance due to a large volume of dead spaces in its nodes when the number of segments in a query trajectory is large. In addition, TB-tree builds a MBR for each segment in a query and performs a range search for each MBR. Because the number of range searches increases in proportion to the number of segments, TB-tree dramatically degrades on trajectory retrieval performance when the number of segments is great.

5 Conclusions

Because moving objects usually moves on spatial networks, instead of on Euclidean spaces, efficient index structures are needed to gain good retrieval performance on their trajectories. However, there has been little research on trajectory access schemes for spatial network databases. Therefore, we proposed an efficient trajectory access scheme for indexing moving objects. For this, we designed our access scheme for efficiently dealing with the trajectories of moving objects on road networks. In addition, we provided an initial trajectory insertion algorithm as well as a segment insertion one, and we provided a retrieval algorithm to find a set of moving objects whose trajectories match the segments of a query trajectory. Finally, we show that our trajectory access scheme achieves about one order of magnitude better retrieval performance than TB-tree. As future work, it is needed to extend our access scheme to a parallel environment so as to achieve better retrieval performance because its parallel execution can be performed easily due to the characteristic of signature files [ZMR98].

References

[B02] T. Brinkhoff, "A Framework for Generating Network-Based Moving Objects," GeoInformatica, Vol. 6, No. 2, pp 153-180, 2002.

[F03] E. Frentzos, "Indexing Objects Moving on Fixed Networks," Proc. of SSTD, pp 289-305, 2003.

[FC84] C. Faloutsos and S. Christodoulakis, "Signature Files: An Access Method for Documents and Its Analytical performance Evaluation," ACM Tran. on Office Information Systems, Vol. 2, No. 4, pp 267-288, 1984.

[HKT02] M. Hadjieleftheriou, G. Kollios, V.J. Tsotras, and D. Gunopulos, "Efficient Indexing of Spatiotemporal Objects," Proc. of EDBT, pp 251-268, 2002.

[PJT00] D. Pfoser, C.S. Jensen, and Y. Theodoridis, "Novel Approach to the Indexing of Moving Object Trajectories," Proc. of VLDB, pp 395-406, 2000.

[PZM03] S. Papadias, J. Zhang, N. Mamoulis, and Y. Tao, "Query Processing in Spatial Network Databases," Proc. of VLDB, pp, 802-813, 2003.

[Se99] S. Shekhar et al., "Spatial Databases - Accomplishments and Research Needs," IEEE Tran. on Knowledge and Data Engineering, Vol. 11, No. 1, pp 45-55, 1999.

[SKS03] C. Shahabi, M.R. Kolahdouzan, M. Sharifzadeh, "A Road Network Embedding Technique for K-Nearest Neighbor Search in Moving Object Databases," GeoInformatica, Vol. 7, No. 3,, pp 255-273, 2003.

[SJK03] L. Speicys, C.S. Jensen, and A. Kligys, "Computational Data Modeling for Network-Constrained Moving Objects," Proc. of ACM GIS, pp 118-125, 2003.

[WMA] http://www.maproom.psu.edu/dcw/

[ZMR98] J. Zobel, A. Moffat, and K. Ramamohanarao, "Inverted Files Versus Signature Files for Text Indexing," ACM Tran. on Database Systems, Vol. 23, No. 4, pp 453-490, 1998.

MECOSIG Adapted to the Design of Distributed GIS

Fabien Pasquasy, François Laplanche, Jean-Christophe Sainte,
and Jean-Paul Donnay

Unit of Geomatics, University of Liège, allée du 6 Août, 17 (B5),
Sart Tilman, 4000 Liège, Belgium
{pasquasy, laplanche, sainte,
donnay}@geomatique-liege.be
http://www.geo.ulg.ac.be

Abstract. For more than ten years MECOSIG has been used as a method for GIS design and implementation in various national and international projects achieved in our laboratory. During a decade, the method has been progressively improved and extended without modification of its basic principles. However the emergence of distributed GIS, implying several organizations capable to play various roles, requires the reappraisal of the methodology. New concerns are identified and a collection of new tools must be deployed. Taking the most of various recent researches completed for public authorities in Belgium, this paper presents some significant adaptations of the original MECOSIG method in order to cope with a distributed GIS environment.

1 Introduction

MECOSIG (*MEthode de COnception de Systèmes d'Information Géographique*) is one of the first methods devoted to GIS design and implementation. Developed in the context of PhD thesis [13] it has been published in 1996 [18]. The method considers a GIS as an information system and, according to the systemic approach, as a component of an organization. The methods dedicated to the design of information systems prevailing at that time (e.g. MERISE in France or SDDADM in U.K.) showed various deficiencies: inflexibility (Cartesian succession of predetermined steps to follow), lack of specific formalisms for spatial data modelling, etc. MECOSIG endeavours to fill in these gaps. It is based on a systemic analysis of the organization concerned by the implementation or the computerization of the GIS and the organization itself is at the core of the method, not only a single database project. Since then this viewpoint is shared by the few published GIS methodological guides [11] [22]. However, MECOSIG is likely the first to cover the complete lifecycle of the GIS and it comes with genuine tools supporting the method which are partially presented in the next paragraphs. The reader interested by the advances and benefits of MECOSIG will find a comprehensive list of references at the end of this paper.

Data, processes and data flows are of course in the list of the main concerns of the method, but the functional structure of the organization and integration controls are added at the same level of concern. During the design process these classes are analysed at different abstraction levels: from the descriptive level to the logical-physical one, corresponding to the implementation as such. The complete design

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1117 – 1126, 2005.
© Springer-Verlag Berlin Heidelberg 2005

protocol can be summarized in a monitoring matrix (table 1) where every cell relates back to specific tools (formalisms, models, prototyping, etc.) and semantic concerns through metadata and documentation. The designer can adopt the way he/she will follow the steps suggested in the matrix according to the special features of the design strategy.

Table 1. The monitoring matrix of MECOSIG

Classes of concern / **Abstraction levels**	*Organization*	*Data*	*Data flows*	*Processes*	*Integration controls*
Descriptive					
Conceptual					
Organizational					
Logical-physical					

Since the first publications dealing with MECOSIG, several improvements and adaptations have been proposed, notably in the framework of graduate and PhD theses. The most significant advances concern the tools available to conceptualize the classes of concern. In this regard it is worth noting that MECOSIG already came with an original and rich formalism for geographical data modelling – so called CONGOO for *CONceptualisation Géographique Orientée Objet* [15] – and a sophisticated typology of topological rules. Improvements have been suggested for instance in order to integrate geographic objects with indeterminate boundaries into conceptual data models [19] and to extend the formalism to spatio-temporal features [21]. Two major advances consisted, on the one hand, to cope with 3D spatial information and relationships [3] [4] and, on the other hand, to introduce UML schemas at different abstraction levels of the design methodology [9]. At the same occasion, the topological rules were simplified and re-formalized. MECOSIG is flexible enough to put up with all these additions and adaptations as long as they do not modify the principles underlying the methodology. In concrete terms, MECOSIG was able to support numerous projects achieved in various public organization (e.g. [14] [16] [17]).

The emergence of distributed spatial databases alters the foundations of information system design. It is particularly true with MECOSIG which puts the organization at the root of the design issue. Nowadays many projects dealing with GIS in Belgium require interoperability and other distributed capabilities, while organizations are involved in spatial data infrastructures at the regional and national levels [5]. Facing with such issues, it was necessary to adapt MECOSIG and to provide new tools fitting the requirements of distributed environments. This paper takes the most of various recent researches that we have completed for public authorities in order to develop common GIS platforms, notably in the framework of the SIGMaTE project [2].

2 Adaptations at the Descriptive Level

Some changes were introduced when compared to [18]. MECOSIG proposed to go twice through the monitoring matrix, first for the analysis phase and secondly for the

conception step. For clearly pragmatic reasons the formal current system analysis and the requirement analysis are henceforth performed at the descriptive level appearing during the phase of conception. It is justified because – practically – the course of the matrix at the analysis level generally resulted in textual descriptions only and hence it generates a waste of time due to the redundancy with the descriptive level of the phase of conception.

It is worth noting that the design of distributed GIS is always based on existing systems which work in a more or less effective way. Distributed GIS implementation is the opportunity to re-conceptualize the spatial databases which have to be distributed. Moreover a common irredundant model of these databases must be proposed on the basis of a single spatial reference framework.

These adaptations affect the classes of concern of the monitoring matrix. The tools used to describe the current situation and to formalize the requirements are the same and they are presented below. This enables to quickly identify the current gaps of the systems and to propose a conceptual solution of the problems in a quite short time.

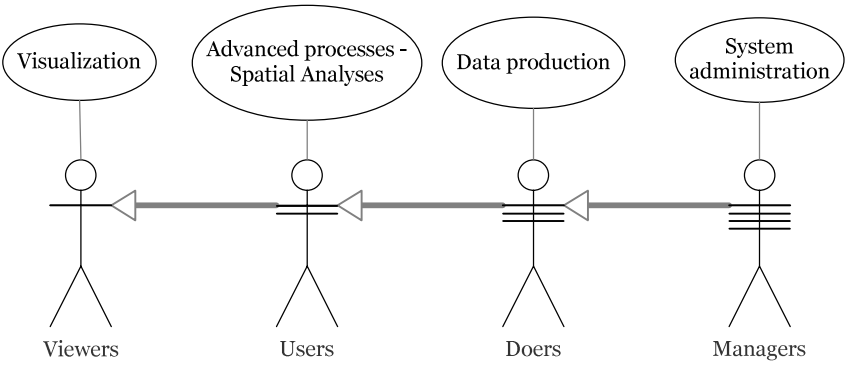


Fig. 1. The four categories of actors and the inheritance of their privileges

In order to describe organization(s) – there could be more than one organization in case of distributed GIS – actors of this or these organization(s) must be identified. These actors are classified in four categories: viewers, users, doers and managers. A set of privileges (figure 1) and specific use cases match every category of actors in the organization(s). Particular symbols can be introduced to distinguish the categories (figure 1). Moreover a specialization relationship exists between these groups. The actors belonging to the higher levels inherit the properties from the lower levels (figure 1). For example, the users inherit the rights granted to the viewers. This classification of actors is not strictly fixed and can evolve according to the cases. Every category can be subdivided to match as well as possible the modeled organization.

The distinction made between doers and managers constitutes an essential characteristic of GIS. Indeed data production and system management are two well differentiated spots which are generally entrusted to different GIS actors in an organization.

The class of concern called "processes" is strongly related to the previously identified actors and in particular crossed processes are specific to distributed GIS. In order to describe them the interactions between the four groups of actors and the system are identified in terms of use cases and they are modeled with UML use case diagrams [12].

Concerning data, there are some changes due to the distributed aspect of the system. Data are still described in a catalogue and their quality must still be analysed. In this respect the use of the standards ISO 19110 [7] and ISO 19115 [8] is suggested. However the observance of standards is not a guarantee of success in a distributed environment as long as data and metadata are concerned. They are essential for specifying the form of the “packaging” but interoperability requires further agreements on the “content” and its meaning. That is in this very place that a significant improvement of the MECOSIG method must be achieved in order to incorporate semantic rules in the spatial data domain. A first step in this direction is to identify data according to the use cases in which they are implied and finally to gather data into batches. Every batch constitutes an intuitive group maintaining a certain number of relations (semantic and/or topological) and intervening together in peculiar activities. Yet the description of the relationship is not mandatory at this level of detail. It will be taken into account at the stage of conceptual modelling. As an example, a specific batch consists in the spatial data of reference which are common to the whole distributed spatial database.

The use cases make an intensive use of data flows. Because the description of the use cases is already performed in the “data” and “processes” classes of concern, the analysis of a specific class of concern devoted to the data flow can be withdrawn in many applications.

3 Adaptations at the Conceptual Level

The UML class diagram is now used as a surrogate for the CONGOO formalism originally proposed with MECOSIG. There are several reasons to explain this transition.

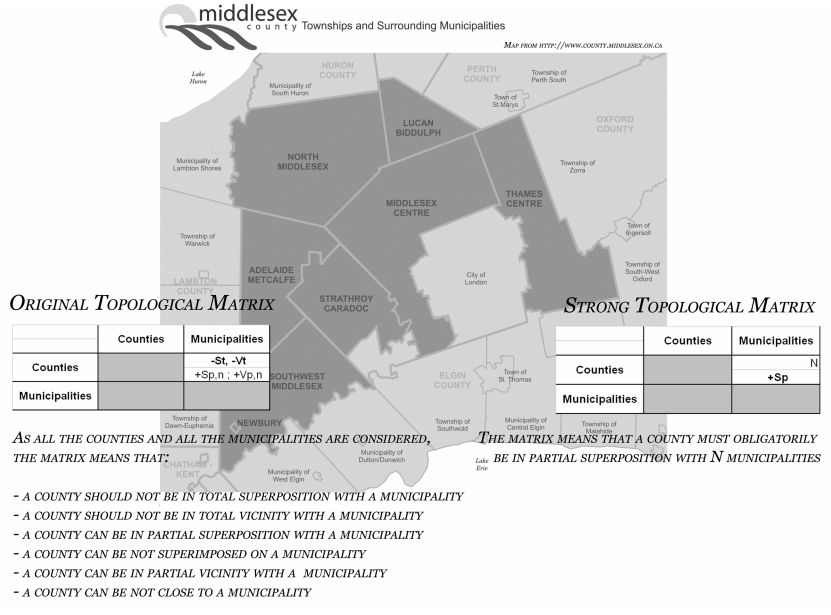
First UML is internationally admitted as standard for system modeling and many case tools are already using it.

Then UML seems to be more suitable for the design of distributed systems. Indeed it offers more than only one diagram to represent the reality and, based on its philosophy, the design can combine more than one model of diagrams to depict a context.

The design is based on the batches identified at the descriptive level. A class diagram is suggested for every batch. That will allow thereafter a more effective distribution of the data between the various spatial databases.

Due to the use of UML instead of CONGOO some specificities of the spatial data design are withdrawn while others become available. The first and probably the most important one is the ability to identify the objects’ geometry. A lot of studies (e.g. [1]) exist to cope with this problem and moreover some mechanisms of extensions are defined in UML. Stereotypes or tagged values permit to adapt UML to the domain of

interest [12]. Another specificity of spatial data is worthy of interest: the management of topological relationships. UML class diagram does not allow the definition of topological relationships like CONGOO does. Moreover these added formalisms can make the model more complex. Introduction of the topological relationships into the model such as suggested in CONGOO is therefore given up. On the other hand, the concept of topological matrix is kept and adapted. One or more topological matrixes can be defined for every data model. Two types of topological matrixes are available: the traditional matrix [18] and the strong topological matrix [9] (figure 2). The latter proposes to define topological constraints having to be respected by all objects belonging to a class.



ORIGINAL TOPOLOGICAL MATRIX

	Counties	Municipalities
Counties		-St, -Vt +Sp,n ; +Vp,n
Municipalities		

STRONG TOPOLOGICAL MATRIX

	Counties	Municipalities
Counties		N +Sp
Municipalities		

AS ALL THE COUNTIES AND ALL THE MUNICIPALITIES ARE CONSIDERED, THE MATRIX MEANS THAT:

- *A COUNTY SHOULD NOT BE IN TOTAL SUPERPOSITION WITH A MUNICIPALITY*
- *A COUNTY SHOULD NOT BE IN TOTAL VICINITY WITH A MUNICIPALITY*
- *A COUNTY CAN BE IN PARTIAL SUPERPOSITION WITH A MUNICIPALITY*
- *A COUNTY CAN BE NOT SUPERIMPOSED ON A MUNICIPALITY*
- *A COUNTY CAN BE IN PARTIAL VICINITY WITH A MUNICIPALITY*
- *A COUNTY CAN BE NOT CLOSE TO A MUNICIPALITY*

THE MATRIX MEANS THAT A COUNTY MUST OBLIGATORILY BE IN PARTIAL SUPERPOSITION WITH N MUNICIPALITIES

Fig. 2. The two kinds of topological matrixes

Concerning the processes design, every use case is extracted from the descriptive level and detailed using UML activity diagrams [12]. The formalization is independent of the implementation. Nevertheless, the activity is replaced within the subsystem of the organization in which it is carried out. The distributed aspect of the system starts to appear in a coarse way. On this level the data catalog can also be enriched by the mention of the activities making use of each data.

Finally sequence diagrams are also used to conceptualize some specific scenarios of activity diagrams [20] (figure 3). It consists in typical sequences describing the interactions between the data and the actors of the organization. The classes of concern "data flow" and "organization" are thus taken into account by the designer.

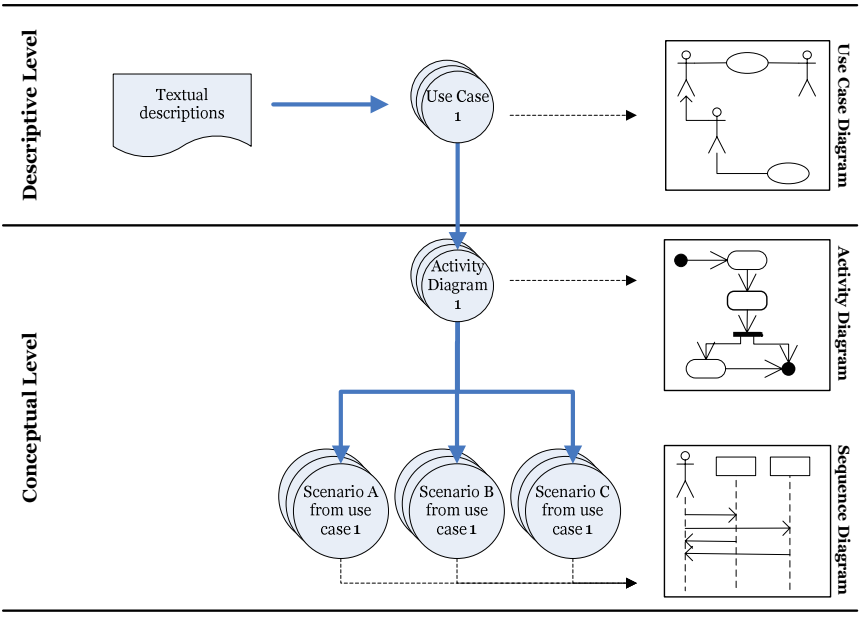


Fig. 3. The tools used at descriptive and conceptual levels

4 Adaptations at the Organizational Level

Facing the implementation of a distributed infrastructure, the components of the system evolve according to their own requirements. From the implementation point of view important organisational constraints must be met. First, the missions of all the actors should be assessed and possibly redefined. Then new tasks are introduced in order to guarantee maintenance, permanent working and growth of the distributed system. Representative committees in charge of coordination tasks will arise for this purpose. These committees will also aim at coordinating actors from the various organizations in-bedded in the distributed system.

Data exchange must be performed easily and in a transparent way for all users of the distributed GIS. The advised use of the data cannot be guaranteed without a free access for all possible users to data about data, i.e. metadata. In this regard again, it is suggested to refer to the ISO standards [7] [8]. Particularly, the metadata related to the quality play a major role in the data usability and in the appraisal of the final results essential in decision-making.

As one of the main objectives of a distributed system is to share data from different organizations, new data flows are generated and others have to be adapted. New applications based on this infrastructure may appear and they can be distributed between all the actors.

Data and/or applications access can be restricted or subject to authorizations. On the other hand acquisition policies for common data could be considered. All these agreements must be considered at the level of the distributed organization and will fall

under the responsibility of the representative committees in charge of the coordination tasks. Some formalization can also be introduced to depict the committees' tasks thanks to the previously proposed tools: UML use case, activity or sequence diagrams according to the steps illustrated on figure 3.

5 Adaptations at the Logical-Physical Level

Because of the numerous actors involved in a distributed system, the roles and tasks of each of them should be clearly specified. The specification and the possible reallocation of the human resources have not to be underestimated because they will condition the implication of the actors in the new system which will be set up. This level offers the opportunity to assess the tasks and the objectives or at the contrary to reorganize the work of each actor. If required, sensitizing and training sessions can be planned and organized in due time in order to facilitate the acceptance and the support for the re-engineered system. A specific attention will be given on the constitution of the coordination task group which must be competent, representative of the various departments or organizations concerned by the distributed spatial system and which must receive the capacity to act decisively. Basically all these tasks are not specific to the implementation of a distributed system however the degree of difficulty is magnified in a distributed framework.

From a technical point of view, some managerial economy and economy of scale could be achieved thanks to the implementation of the distributed system. The acquisition of common reference data and the globalisation of software licence costs are two examples between many others. However the objective of the distributed operation is certainly not the standardization of the hardware and software resources. GISs, DBMSs and data/application servers likely will remain in the different parts of the distributed organization, in order to avoid a complete disrupt in the respective businesses. The technical analysis which must be completed at the physical level concerns the add-on hardware/software needed to guarantee the required degree of interoperability specified during the design analysis. The main reason of a technical study is to identify precisely the distributed system components but also to assess the capability of the whole to provide the required level of service. The ability of a user to access consistently and coherently spatial data and processes across distributed databases depends on semantic interoperability [6]. Providing systems for cross-relating items of information across multiple sources is needed to solve problems.

As early mentioned, UML class diagrams were used to build up a conceptual model allowing to note classes, attributes and associations needed to describe the distributed databases structure. The modelling process gives place to the design of many objects which must be deployed in a database environment and that claims many decisions concerning the logical-physical level [10]. In this connection, the design aspects are mainly related to the data storage structure. The design of the data storage and the knowledge of their physical location on servers, discs (even of multiple partitions) will facilitate the final management of the databases. Using UML to model tabelspaces, to specify the discs containing them and their physical location are very important tasks which must be planned carefully and precisely. Components and deployment UML diagrams offer objects well suited to describe all these

functions and of course to expose them graphically. For instance (figure 4), one database of the distributed system is designed and is related to various tabelspaces and users by some dependence relationships.

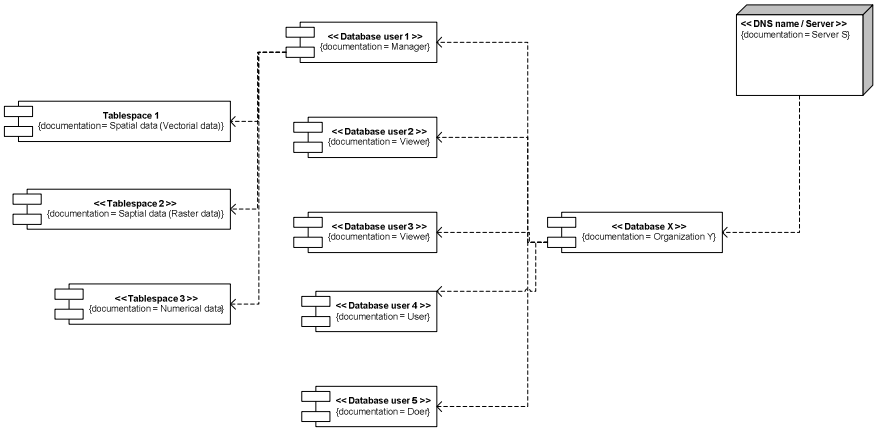


Fig. 4. Components diagram (UML) designing data storage structure

Activity and sequence UML diagrams produced at the conceptual level can also be improved by adding objects dealing with data flows, procedures and process according to GIS software, tools and DBMS specificities which will be used in the distributed system.

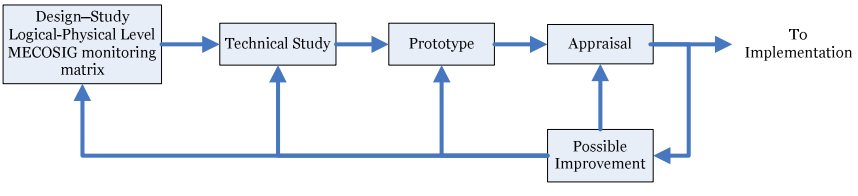


Fig. 5. The evolutionary project pilot

Lastly and recommended, whole of the logical-physical level is integrated in an evolutionary project pilot. A technical study of the numerous software components from the identified infrastructure will be considered. A prototype will be implemented as well on the basis of elements previously identified (figure 5).

In this connection, it is still necessary to find out the performing level of networks, DBMS and spatial data servers, GIS software, etc, in order to guarantee the implementation of an appropriate distributed system. The interoperability level required (data interoperability and/or GIS services interoperability) is decisive in the technical choice of distributed system components.

6 Conclusions

Distributed GIS and spatial data infrastructures have the special feature of being made over existing GISs and DBMSs. This does not imply that the design analysis of the future distributed system can be avoided. On the contrary, this kind of project requires a sound study from the conceptual to the physical level. MECOSIG proved to be flexible enough to support such design analysis provided that it is assisted by suited tools taking place in a revisited monitoring matrix. This paper shows that several diagrams issued from UML can offer interesting capabilities in the field.

Based on actual experiences requiring a comprehensive level of interoperability between distributed data, this paper is deliberately vague about the possible implementation of distributed processes. The tools dedicated to the processes (formal description…) presented in this paper could provide some elements in this direction. However a comprehensive review of this topic would require further experiences embedding likely a stronger dependency on software capabilities.

References

1. Bédard, Y, S. Larrivée, M.J. Proulx, M. Nadeau: Modeling Geospatial Databases with Plug-Ins for Visual Languages: A Pragmatic Approach and the Impacts of 16 Years of Research and Experimentations on Perceptory, S. Wang et al. (Eds.): COMOGIS Workshops ER2004, LNCS 3289, (2004) pp. 17–30.
2. Bels F, Capoen E., Pasquasy F., Swennen C.: Projet d'assistance technique, méthodologique et scientifique dans le domaine de la cartographie et des S.I.G., en vue d'implémenter un prototype d'infrastructure d'information géographique répartie entre la DGATLP et la DGRNE du Ministère de la Région wallonne, Final report, unpublished, (2003) 310 p.
3. Billen R.: Introduction of 3D information in urban GIS: a conceptual view. International Archives of Photogrammetry and Remote Sensing, Vol. XXXIII, part B3, Amsterdam (2000), pp. 73-78.
4. Billen R., Zlatanova S.: 3D Spatial relationships model: a useful concept for 3D cadastre?, Proceedings of the International workshop on "3D Cadastre", Delft University, (2001) pp. 1-17.
5. Donnay J.-P.: Distributed GIS for sharing large scale data between public agencies. A case study in Belgium. GIS 2002 International Conference Proceedings, The Bahrain Society of Engineers, Bahrain, (2002). pp. 47-59.
6. Fonseca F., Egenhofer M.: Ontology-Driven Geographic Information System, 7th ACM Symposium on Advances in Geographic Information Systems, Kansas City, MO, C. Bauzer Medeiros (ed.), (1999)
7. ISO/TC211 19110: Geographic Information: Methodology for feature cataloguing (2004)
8. ISO/TC211 19115: Geographic Information: Metadata (2003) 140 p.
9. Laplanche F.: Conception de Projet SIG avec UML. Bulletin de la Société géographique de Liège, n° 42, (2002) pp. 19-25.
10. Naiburg E., Maksimchuk R.: Bases de données avec UML. Collection Référence, CampusPress, Gap, (2002) 293 p.
11. New York State Archives, Calkins H.: Geographic Information Systems Development Guides (1996)
12. http://www.archives.nysed.gov/a/nysaservices/ns_mgr_active_gisguides.shtml
13. Object Management Group: UML Specification version 2. http://www.uml.org. (2004)

14. Pantazis D.: Analyse méthodologique des phases de conception et de développement d'un système d'information géographique. PhD thesis unpublished. University of Liège. (1994)
15. Pantazis D.: Le développement de la base de données géographiques du Ministère de l'Aménagement du Territoire du Grand Duché de Luxembourg, actes du colloque Les systèmes d'Information en Géographie, Org.: Université de Genève, Université de Fribourg, Chexbres, Suisse, (1995)
16. Pantazis D.: CON.G.O.O.: A conceptual formalism for geographic database design, Geographic Information Research, Bridging the Atlantic, Ed. M. Craglia & H. Coucledis, Taylor & Francis, (1997) 348-367.
17. Pantazis N. D., Cornélis B.: Designing and implementing a GIS in an international context, Transactions in GIS, vol. 1, n° 4, (1997) 301-320
18. Pantazis D., Cornélis B., Billen R., Sheeren D.: Establishment of a geographic data dictionary: case study on the Brussels regional government GIS. Computer, Environment & urban Systems. 26(1), (2002) pp. 3-17
19. Pantazis D., Donnay J.P.: Conception de S.I.G.: Méthode et formalisme. Collection Géomatique, Hermes, Paris, (1996) 352 p.
20. Pantazis D. & Donnay J-P.: Objets géographiques à limites indéterminées. Modélisation et intégration dans un modèle conceptuel de données. Revue internationale de géomatique, 7(2), Hermes, (1998) pp. 159-186.
21. Roques P.: UML par la pratique: étude de cas et exercices corrigés. Eyrolles, Marsat. (2001) 302 p.
22. Sheeren D.: La conception de bases de données spatio-temporelles, problématiques et solutions dans le cadre du formalisme CONGOO, Graduate thesis unpublished, University of Liège, (1999) 113 p.
23. Somers R.: Developing GIS Management Strategies for an Organization. Journal of Housing Research, Vol.9, n° 1, (1998) pp. 57-77

The Emerge of Semantic Geoportals

Athanasis Nikolaos, Kalabokidis Kostas, Vaitis Michail, and Soulakellis Nikolaos

Department of Geography, University of the Aegean,
University Hill, GR-811 00 Mytilene, Greece
athanasis@geo.aegean.gr, kalabokidis@aegean.gr,
vaitis@aegean.gr, soulakellis@aegean.gr

Abstract. Geoportals (geographic portals) are entry points on the Web, where various geographic information resources can be easily discovered. They organize geospatial data and services through catalogs containing metadata records, which can be queried in order to give access to related resources. However, the current organization of geographic information in metadata catalogs is unable to capture the semantics of the data being described; therefore users often miss geographical resources of interest when searching for geospatial data in the World Wide Web. In this paper, we present an innovative approach in the development of geoportals, based on the next generation of the Web, called the Semantic Web. This approach relies in the organization of the geo-data at the semantic level through appropriate geographic ontologies, and the exploitation of this organization through the user interface of the geoportal. To the best of our knowledge, this approach is the first that combines the expressiveness of geo-ontologies in the context of geographic portals.

1 Introduction

Geographic Information Systems (GIS) and spatial databases provide nowadays the necessary management, analysis and visualization of geographic knowledge. The revolution in network connectivity and online availability has helped in the access of geospatial resources via the World Wide Web [14]. However, many geographical data are scattered among different agencies and organizations. There is a plethora and diversity of geo-data standards, formats and terminology representing geospatial concepts, that encumbers the sharing, dissemination and exploitation of this heterogeneous geographical information [8]. As the demand for geospatial data increases, the lack of interoperability in the area of Geographic Science becomes critical. This heterogeneity that enhances the lack of interoperability is based on [10]:

- Syntactic reasons (*syntactic heterogeneity*). Syntactic heterogeneity is related with the differences among the different Data Base Management Systems (i.e. relational vs. Object Oriented databases). This kind of heterogeneity is also related with the different representations of geographic objects (i.e., raster vs. vector data).
- Schematic reasons (*schematic heterogeneity*). Schematic heterogeneity appears when for the same geospatial phenomena different generalization/ junction hierarchies are used.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1127 – 1136, 2005.
© Springer-Verlag Berlin Heidelberg 2005

- Semantic reasons (*semantic heterogeneity*). This kind of heterogeneity is the most significant source of implications in geographical data exchange. It arises from the different ways in modeling and perception of the same or related data. This lack of common understanding of the same data is also known as cognitive heterogeneity. Other semantic conflicts arise for homonyms (same word for different concepts) and synonyms (same concept is described by different names).

The Open Geospatial Consortium (OGC)[1] has already addressed some basic issues about GIS interoperability. OGC is an association of software developers, and government agencies that aims to define a set of requirements, standards and specifications in order to achieve geographic information reuse and discovery. The Geographic Markup Language (GML)[2], developed by OGC, provides a syntactic approach to encode geospatial information. GML is the most common way to geographical data exchange across different GIS applications and platforms. However, it is unable to cope with discrepancies in the meaning (semantics), interpretation and intended use of geographic concepts, thus conducing to a deficient exploitation of the data exchanged. Therefore, there is a vital need of a more efficient mechanism for discovery of required geo-data that overcomes semantic heterogeneity issues. Today's retrieval methods offer little support for any deeper structures that might lie hidden in the data; therefore users may often miss critical information when searching the World Wide Web [5]. The idea of a Semantic Web introduced by Tim Berners-Lee [2] proposes "a web of data that can be processed directly or indirectly by machines", bringing a higher degree of automation in exploiting data in a meaningful way. Semantics is captured by associating formal descriptions to provide well defined meaning to data and other Web resources, so that information processing can be based on meaning. Current efforts of the Semantic Web include the Resource Description Framework (RDF), Topic Maps, and the Web Ontology Language (OWL).

In this context, the advent of the Semantic Geospatial Web promises better exploitation methods of geographic information by incorporating the data's semantics and benefit from them during the search process [7]. Dominant role in the emerging Geospatial Semantic Web hold geo-ontologies, which are vocabularies of geographic terms that provide the semantics of data and define a set of domain concepts and their relationships.

Thus, for the accomplishment of the Geospatial Semantic Web, there is a need for:

- Development of spatial ontologies each with a formal semantics;
- The representation of those semantics such that they are available both to machines for processing and to people for understanding;
- The processing of geospatial queries against those geographic ontologies and the evaluation of the retrieval results.

As the demand for a better access to geographic information in the World Wide Web increases, many Web sites provide navigation and searching for geo-data and services. Geographical Portals, also called geoportals, give online access to collections of geospatial data. They are single points of access to geographic datasets, GIS applications, static/interactive maps, map layers etc.

[1] http://www.opengeospatial.org
[2] http://www.opengis.net/gml/

The organization of geographic resources in geoportals is achieved through metadata records. These metadata records are collected into a comprehensive metadata catalog that can be indexed by various means (i.e., geographic location, time etc.). It is important to mention that geoportals are not used to store the geospatial data. On the contrary, they provide the organization of the geospatial data through the metadata records, as well as links (hyperlinks) to the data itself.

However, the current organization of geographic information in metadata catalogs has some important implications, which degrade the efficient searching of geographic resources in geoportals. First, the organization is unable to capture the semantics of the data being described, and second, geospatial metadata that come from different geospatial metadata standards cannot be uniformly exploited. This paper describes an innovative approach in the implementation of geoportal's metadata catalogs. This approach is based on the semantic organization of the various resources through geo-ontologies, and the exploitation of this organization through the user interface of the geoportal.

The remainder of this paper is organized as follows; Section 2 introduces the main concepts relative to geographical portals and Section 3 describes the proposed semantic way to the implementation of metadata catalogs, as well as the benefits of this organization of geospatial metadata. Finally, Section 4 concludes our paper and discusses possible future research directions.

2 Geoportals

Geographical portals are simply gateways to geographic resources. They are web environments that allow a community of users to discover and share geospatial content and services [13]. The origins of geographical portals can be traced to the early growth of the World Wide Web. Web sites like MapQuest[3] or MapBlast[4] capitalized the interest of the research community to locate and map places on the World Wide Web [14]. As the Web matured, many organizations started to handle geographic information, like the Federal Geographic Data Committee (FGDC)[5], and created web sites known as clearinghouses, that allowed to search for geo-data through keywords or to navigate across organized thematic categories of geographic information.

Geoportals can be subdivided into two main categories: catalog geoportals and application (services) geoportals. Catalog geoportals primarily organize access to geographical information. Representative examples of catalog geoportals are the Geospatial One Stop[6] geoportal which is the extension of the NSDI clearinghouse, the geo data portal[7], and the G-Portal [4]. Application (services) portals provide on-line dynamic services (for example, MapQuest provides routing services, the GIS Portal Toolkit provides mapping services etc.).

Independently of what main category they belong, almost all geoportals offer the same main features and services:

[3] http://www.mapquest.com/
[4] http://www.mapblast.com/
[5] http://www.fgdc.gov/
[6] http://geo-one-stop.gov
[7] http://geodata.grid.unep.ch

- Searching for geographic data: These geo-data could be anything from static or interactive maps to GIS services, 3-D scenes or geographic datasets, etc. Searching for geospatial resources in geoportals is achieved either by following hyperlinks between thematic categories in which the geospatial resources are categorized (navigation), or by providing spatial, thematic, temporal, or keywords search criteria (querying).
- Describing the geospatial resources with metadata: A prominent feature of all geoportals is a catalog service for publishing and accessing metadata. In order to effectively disseminate knowledge, geo-portals have catalogs holding descriptions (i.e., metadata) about the available resources. Geospatial metadata give answers to when, where, what or who about the information they describe. Metadata are collected and recorded in comprehensive metadata catalogs [14]. These catalogs can then be searched through the portal to find related information about geographical datasets or other data holdings. Geoportals just provide the organization (metadata records) and the means to find desirable resources; they do not store the data itself. Access to the resources is given through on-line links to the data. Many geoportals also allow the incorporation of new sets of metadata by allowing its users to publish their metadata in the geoportal. Thus, the information sets of the geoportal are dynamic and the portal is enriched all the time by its own users.

3 A Semantic-Based Organization of Geospatial Metadata

However, there are some significant implications in the current information organization of geoportal's metadata catalogs:

- There is a lack of a uniform framework and standard for describing geographic metadata. Till today, many metadata standards have been arise, each with its own particularities. These standards provide terms describing elements common to most geospatial data and encourage people who document geographical information to use these terms. In the Content Standard for Digital Geospatial Metadata-CSDGM)[8], the metadata are represented in a hierarchical form in plain ASCII files, where each line contains one element (from the total of 119 that the standard contains) as well as the value of this element. Table 1 shows a snippet of CSDGM metadata elements about the UTM coordinate system.

Table 1. A snippet of metadata elements in CSDGM metadata standard

```
Universal_Transverse_Mercator:
  UTM_Zone_Number: 1
  Transverse_Mercator:
    Scale_Factor_at_Central_Meridian: 0.9996
    Longitude_of_Central_Meridian: -177
    Latitude_of_Projection_Origin: 0.0
    False_Easting: 500000
    False_Northing: 0.0
```

[8] http://www.fgdc.gov/metadata/contstan.html

Another favorite geospatial metadata standard is the ISO 19115 standard[9] released from the ISO Technical Committee 211[10]. This standard defines more than 300 metadata elements in the area of geographical information / geomatics. It is fairly obvious that this diversity of geospatial metadata standards also means that there is a difficulty to describe geo-data with metadata coming from different standards. Thus, the uniform exploitation of geographical information in the geoportal is difficult to be achieved. What is needed is a common framework that will be able to describe uniformly the geospatial resources and will allow the integration of geographic data in an interoperable way.

- The current metadata standards for describing geographical information are not able to capture the semantics of the data they describe. Both in the CSDGM and the ISO 19115, the representation of the metadata is based on the XML data format. CSDGM uses the Standard Generalized Markup Language- SGML11, and tools like 'Xtme' (Xt metadata editor)12 and 'mp' (metadata parser)13 are used for the corresponding editing and parsing of the metadata. In the ISO 19115 a pure XML representation is followed, and each metadata file conforms to an appropriate XML Schema14. Even geoportals that use none of these metadata standards, like G-Portal, follow a similar XML-based way for describing their metadata. Nevertheless, despite the great popularity of XML, it cannot be used for an automated and efficient management of the geospatial information of geo-portals. The interoperability that XML provides is restricted from different meanings that anyone can give to the geographical information metadata [5].

What we propose is an ontology-based organization of geospatial metadata. With the use of ontologies, we provide a semantically meaningful way for the creation and exploitation of the geo-portal's resources. The advantages of generating geo-ontologies are [6]:

- Geo-ontologies can play the role of a common vocabulary describing different geospatial resources;
- Geo-ontologies can be used to help the geo-data providers to enter the metadata in a semantically valid form;
- Using shared geo-ontologies the interoperability between heterogeneous geographic resources can be achieved.

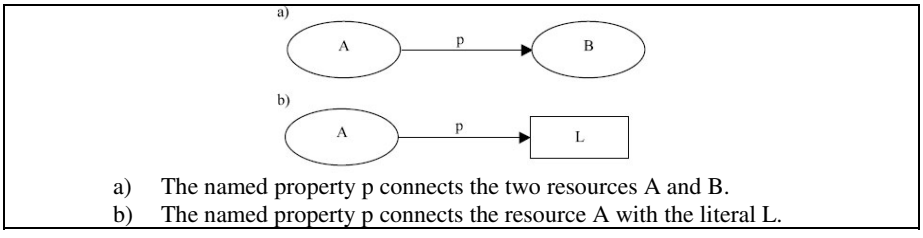
More specifically we propose the exploitation of the Resource Description Framework (RDF) [12] and its corresponding schema specification (RDFS) [3] as the appropriate data model for the management (encoding, exchange and process) of geographic metadata. More precisely RDF provides:

- a Standard Representation Language for metadata based on directed labeled graphs, in which nodes are called resources (or literals) and edges are called properties;

[9] http://www.standardzworld.com/iso-19115.htm
[10] http://www.isotc211.org/
[11] http://www.w3.org/MarkUp/SGML/
[12] http://geology.usgs.gov/tools/metadata/tools/doc/xtme.html
[13] http://geology.usgs.gov/tools/metadata/tools/doc/mp.html
[14] http://metadata.dgiwg.org/ISO19115/ISO19115_v0_5_detail.htm

- a Schema Definition Language (RDFS) for creating vocabularies of labels for these graph nodes called classes and edges called property types; and
- a XML syntax for expressing metadata and schemas in a form that is both humanly readable and machine understandable.

In RDF, resources are organized in descriptions (resource descriptions), represented as directed labeled graphs. These graphs are also called nodes and arcs diagrams. Arcs represent properties. Each property connects two nodes, coming from a node representing a resource (drawn as oval) and pointing to another resource or a literal (drawn as rectangle). So we have the following two options as shown in Figure 1.



a) The named property p connects the two resources A and B.
b) The named property p connects the resource A with the literal L.

Fig. 1. Resource descriptions in RDF

The RDF data model specification provides no mechanism for declaring properties, nor does it provide any mechanisms for defining the relationships between properties and other resources; that is the role of the RDF schema. RDF schemas are also represented as directed acyclic labelled graphs and are essentially vocabularies of labels for graph nodes called classes or literals and edges called property types (Fig. 1).

Our approach is based mainly on the organization of metadata that provides RDF. We suggest to organize the geographic resources in resource descriptions and to provide their corresponding RDF Schemas. Figure 2 illustrates how we propose to use RDF in order to provide a semantic meaningful geo-portal's metadata catalog.

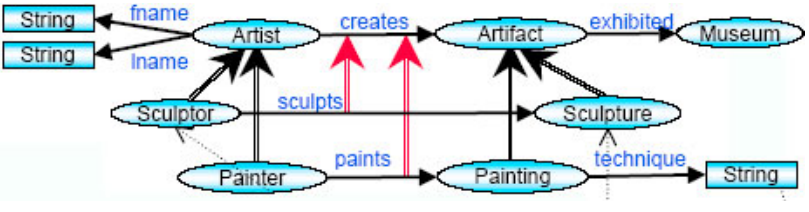


Fig. 2. The directed acyclic graph of a RDF Schema

The metadata catalog of Figure 2 is for a hypothetical geoportal that is specified in representing geographic resources about natural disasters. The resources are described according to two RDF Schemata: The first one is based on the ADEPT

geo-ontologies[15], while the second is a hypothetical RDF schema that supports some of the core metadata of the Content Standard for Digital Geospatial Metadata (CSDGM). Each one of these RDF Schemata describes the geographical resources of the geoportal under a different spectrum. In the first one the emphasis is given in the taxonomy and relationships of natural disasters. It contains a general class Natural Disaster, with more specific classes Earthquake and Volcano. Class Earthquake is associated with class Volcano through property type causes, while class Natural Disaster is associated with class Geographic Entity through property type affects. In the second RDF Schema emphasis is given in metadata that are concerned about core geographic attributes of the resources (i.e longitude/ latitude), as well as online access to the corresponding resources.

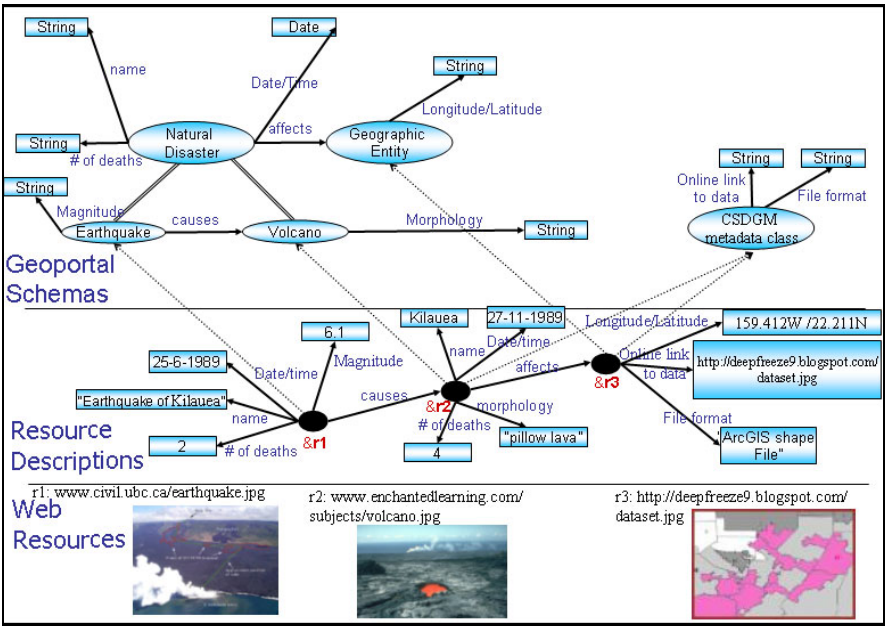


Fig. 3. A semantic-based organization of geoportal's metadata catalog

Our hypothetical geo-portal contains geographical metadata about three available resources: an earthquake of Kilauea, Hawaii (identified as r1), a volcano (identified as r2), and a geographic entity (identified as r3). Resource r1 has its own magnitude and resource r2 has its own morphology. Both inherit properties '# of deaths', name and date/time from the superclass Natural Disaster. Resource r3 is classified under the class Geographic Entity and has a specific Longitude/ Latitude. At the same time, resource r3 has additional information like 'file format' and 'on-line link to data'. These attributes are acquired from the 'CSDGM metadata class' which belongs to the other geo-ontology. Thus, our approach allows the exploitation of different

[15] http://www.alexandria.ucsb.edu/historical/www.alexanria.ucsb.edu/adept/proposal.pdf

geo-ontologies (RDF Schemata), each describing the geographical metadata with its own set of metadata.

It is obvious that this ontological representation of the geospatial metadata provides the means to capture the semantics of the data being described. Furthermore, in contrast with geoportals already implemented, we suggest to exploit the semantics of geographical metadata with the help of semantic-based query languages, in order to provide the means to search for geographical resources of interest. In our approach we use the Resource Description Framework Query Language (RQL) [9]. RQL adapts the functionality of semistructed / XML query languages to the peculiarities of RDF, but foremost, it enables to uniform query both resource descriptions and schemas. In addition, we believe that semantic query languages such as RQL are mainly targeting experienced users, who need to understand not only the RDF/S data model but also the syntax and semantics of the query language in order to formulate a query.

We propose to exploit the expressiveness of RQL in a transparent way, while the user is navigating and browsing through the geoportal, a method based on the context of semantic web portals [11]. At this context, users navigate through semantic hyperlinks capturing the conceptual relationships of different resources. In other words, the semantic relationships between the different classes of the geo-ontologies are represented as hyperlinks, helping the users of the geoportal to navigate in a semantic way through the portal. While users are browsing, corresponding RQL path expressions are generated which capture accurately the meaning of its navigation steps. Additionally, at each navigation step users can enrich the generated queries with filtering conditions on the attributes of the currently visited class while they can easily specify which class of resources should be finally included in the query result. Thus, with the use of semantic-based query languages, searching for geospatial data becomes more efficient, because the resources found match to the corresponding navigation steps in the graphical user interface of the geoportal. Figure 4 is a snapshot of the semantic web portal of GRQL.

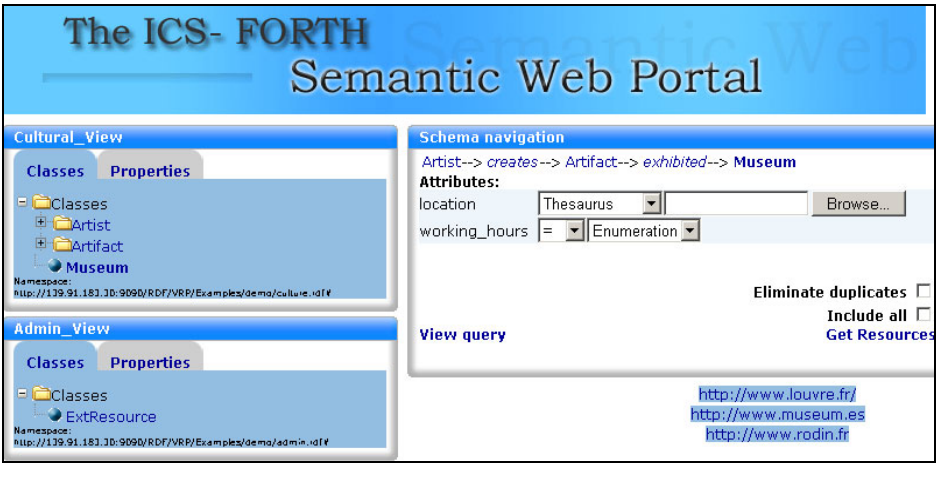


Fig. 4. The Semantic Web portal of GRQL

For a better understanding of the potential of our approach, the following navigation scenario (based on the metadata catalog of Figure 3) is given: The user navigates in the geoportal and searches for earthquake resources of interest. The user visits first the page that contains information about 'Natural Disaster' in general, and follows the hyperlink to the page about 'Earthquakes'. From there, and according to the schema of Figure 1, users visit the page about Volcanoes because there is a semantic relation between these two concepts, provided by a hyperlink by the geoportal. After this, and in a similar way, users find the corresponding affected geographic entities by visiting the class Geographic Entity. With his navigation completed, the geoportal provides a much richer set of information, according to the navigation steps followed.

Thus, users followed the navigation steps above could easily find, for example:

- Earthquakes that caused volcanoes in the last century;
- Shape files of volcanoes, as well as the earthquakes which caused them.

In a summary, the advantages of exploiting a semantic web-based organization in the metadata catalog of geoportals are fairly obvious:

- The proposed organization of geoportal's metadata catalogs provides the means to capture the semantics of the data being described;
- It allows the uniform integration and exploitation of different geo-ontologies (i.e., RDF Schemata), each describing the geographical metadata with its own set of metadata;
- The exploitation of semantic query languages leads to a more efficient searching for geospatial data in the geoportal.

4 Conclusions and Future Work

In this paper, we presented an innovative approach in the development of geo-portals, based on the next generation of the Web, called the Semantic Web. This approach relies in the organization of the geographic resources at the semantic level through appropriate geographic ontologies, and the exploitation of this organization through the user interface of the geoportal. Up to date, the data models used for geospatial metadata are unable to capture the conceptual meaning of the data being described. By providing a semantic rich organization, geoportals will not only solve the semantic heterogeneity problem, but will also support higher quality and more relevant information resources. Furthermore, with the exploitation of declarative query languages, like the RDF Query Language (RQL), geoportals will be able to provide high-level mechanisms to geographical resources, according to semantic web models like RDF/S.

Currently, we are implementing a semantic-based geoportal specialized in information on natural disasters. Simultaneously, we are developing the geo-ontologies (RDF Schemata) as well as the corresponding resource descriptions that will be used as semantic-based metadata catalog of the geoportal. As a future goal, we plan to extend this approach to support publishing new metadata through the graphical user interface of the geoportal.

References

1. Athanasis, N., Christophides, V. and Kotzinos, D., Generating On the Fly Queries for the Semantic Web: The ICS-FORTH Graphical RQL Interface (GRQL), Proceedings of the 3rd International Semantic Web Conference (ISWC'04), Hiroshima, Japan, 2004.
2. Berners-Lee, J., Hendler, J. and Lassila, O., The Semantic Web, Scientific American, vol. 184, no. 5, pp. 34-43, 2001.
3. Brickley, D. and Guha, R.V., Resource Description Framework (RDF) Schema Specification. Proposed Recommendation, 1999.
4. Chang, C.-H., Hedberg, J. G., Theng, Y.-L., Lim, E.-P., The, T.-S., Goh, D. H.-L., Evaluating G-Portal for Geography Learning and Teaching, Proceedings of the Joint Conference on Digital Libraries (JCDL2005), Denver, Colorado, June 2005.
5. Christophides, V., Plexousakis, D., Scholl, M., Tannen, V., The Semantic Web: Myths and Reality, The Onassis Foundation Lectures Series in Computer Science: Internet and Web: Crawling the Algorithmic Foundations, 2003.
6. Egenhofer, M. J., Toward the Semantic Geospatial Web. Proceedings of the 10th ACM Intl' Symposium on Advances in GIS, McLean, Virginia, 2002.
7. Fonseca, F., Seth, A., The Geospatial Semantic Web, UCGIS Research Priorities, 2002.
8. Fonseca, F., Egenhofer, M., Davis, C. and Borges, K., Ontologies and Knowledge Sharing in Urban GIS. Computer, Environment and Urban Systems, 2000.
9. Karvounarakis, G., Magkanaraki A., Alexaki, S., Christophides, V., Plexousakis, D., Scholl, M., Tolle, K. RQL: A Functional Query Language for RDF, The Functional Approach to Data Management: Modelling, Analyzing and Integrating Heterogeneous Data, LNCS Series, Springer-Verlag, 2004.
10. Kokla, M., Kavouras, M., Theories of Concepts in Resolving Semantic Heterogeneities, Proceedings of the 5th AGILE Conference on Geographic Information Science, Palma de Mallorca, Spain, April 2002.
11. Kotzinos, D., Pediaditaki, S., Apostolidis, A., Athanasis, N. and Christophides, V., Online Curriculum on the Semantic Web: The CSD-UoC Portal for Peer-to-peer e-learning, Proceedings of the 14th Intl' World Wide Web Conference (WWW'05), Chiba, Japan, 2005.
12. Lassila, O., Swick, R., Resource Description Framework (RDF) Model and Syntax Specification, 2001.
13. Maguire, D. J., Longley, P. A., The emergence of geoportals and their role in spatial data infrastructures, Proceeding of the 7th Conference on Global Spatial Data Infrastucture, 2004.
14. Tait, M. G., Implementing geoportals: applications of distributed GIS. Computers, Environment and Urban Systems 29(1), 2005.

Ontology Assisted Decision Making – A Case Study in Trip Planning for Tourism

Eleni Tomai[1], Maria Spanaki[1,3], Poulicos Prastacos[2,3], and Marinos Kavouras[1]

[1] National Technical University of Athens, Cartography Laboratory,
15780 Zografos Campus, Greece
{mkav, spanaki, etomai}@mail.ntua.gr
http://ontogeo.ntua.gr
[2] FORTH, Institute of Applied and Computational Mathematics,
Heraklion, Greece
poulicos@iacm.forth.gr
http://www.iacm.forth.gr/regional
[3] InfoCharta Ltd
http://www.infocharta.gr

Abstract. Traditional trip planning involves decisions made by tourists in order to explore an environment, such as a geographic area, usually without having any prior knowledge or experience with it. Contemporary technological development has facilitated not only human mobility but also has set the path for various applications to assist tourists in way-finding, event notification using location-based services etc. Our approach explores how the use of ontologies can assist tourists plan their trip, in a web-based environment. The methodology consists of building two separate ontologies, one for the users profile and another one concerning tourism information and data in order to assist visitors of an area plan their visit.

1 Introduction

This paper addresses the issue of trip planning in the context of web services. Tourists present a special category of agents since they are on the move, they are very different from each other, have diverse interests and more importantly they are eager to explore an area for which we assume they have little prior information or knowledge.

Several approaches have been presented, most of which making use of location-based services and mobile technologies, which provide services for tourists. In [9] and [10], the need for user profiles in location-based services is explored. While in [8] the use of a context-aware system integrated in a mobile application is proposed for assisting tourists. Finally, in [5] a mobile system is introduced that offers guided tours using a semantic matching algorithm.

The system we propose herein is governed by the following concepts: since tourisms are not a group with homogeneous characteristics, the notion of *personalization* is crucial in the design of a decision support web service that helps them plan a trip. In [2] the development of Personalized Information Systems in a web environment is discussed in order to handle the plethora of available data on the web. Another impor-

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1137 – 1146, 2005.
© Springer-Verlag Berlin Heidelberg 2005

tant issue is that of *context*, referring to the usability/ conformity of the system's answer to the user as a result.

To be more specific we propose a web service that can answer to the following types of questions:

- I have two days to spend in X, what do you propose me to do?
- Today I want to do some sightseeing in X and then go to sea.

In order to provide an answer, the system should include a conceptual model of the user profile. This is achieved by presenting to the user a questionnaire through a web based interface, so that the user's personal information, preferences, needs and interests can be extracted and recorded in a user profile ontology. The other dimension of the system is the tourism ontology that contains actual information on a specific area of interest. We have created a case study ontology for the prefecture of Heraklion, in the island of Crete, that we present herein. Although it is applied to Heraklion, a similar ontology can be applied elsewhere as well.

The remainder of the paper is constructed as follows: section 2 roughly sketches the system architecture. While section 3 details the user profile ontology along with the user interface, and section 4 presents the ontology concerning tourism information. The context matching algorithm, which generates the mapping between the above mentioned ontologies, is explained in section 5. Finally section 6 demonstrates further research challenges.

2 System Architecture

This section presents the components of the system. These consist of the two ontologies, namely user profile and tourism ontology, the web-based user interface, the context matching algorithm and the map server.

Starting with the two ontologies, their main difference is that the user profile ontology is elicited by the users' responses to the interface. However this procedure is not entirely free. On the contrary, this is done according to a predefined generic ontology, which facilitates the elicitation process and guides – to a certain extent- the personalization of the system. Furthermore, the user profile ontology gets populated as more users utilize the system. The predefined ontology will be thoroughly presented and explained in section 3. On the other hand, the tourist ontology is populated in advance by the service provider, with real data, and only when he/she wants to update/expand the included information he/she can add more instances to the ontology. The main dimensions of the tourism ontology and how data is organized therein are explained in section 4.

Apart from the two ontologies, in direct contact with the user is the interface for eliciting the user's characteristics. The interface poses ontology-driven queries to elicit information concerning the user. The terminology used in the interface is in accordance with the terminology used in the user profile ontology (non-populated at first). The answers of the user are recorded by the system and included in the user profile ontology as its instance that has properties (characteristics) with specific values. A more detailed presentation of the interface can also be found in section 3.

Another important component of the system is a map server which shows the location of the tourism ontology's concepts of interest. In addition, the map server is utilized to visualize the answer of the system, so that the proposed places and itineraries are portrayed to the user.

The system provides an answer to the query of the user using a context matching algorithm which matches the user profile to the tourism ontology, so that the answer given, matches user needs and interests. The detailed functions of the algorithm are presented in section 5. The characteristics of the algorithm and its ability to generate mappings between the two ontologies, guarantee the conformity of the answer.

The systems works in two steps: first the user fills in the interface so that his/her profile is generated, second the user states the question. Then, the system runs the context matching algorithm between the two ontologies and returns the answer as a text but also locating the proposed places/ points of interest on the map (fig. 1 shows the system architecture, and the procedure).

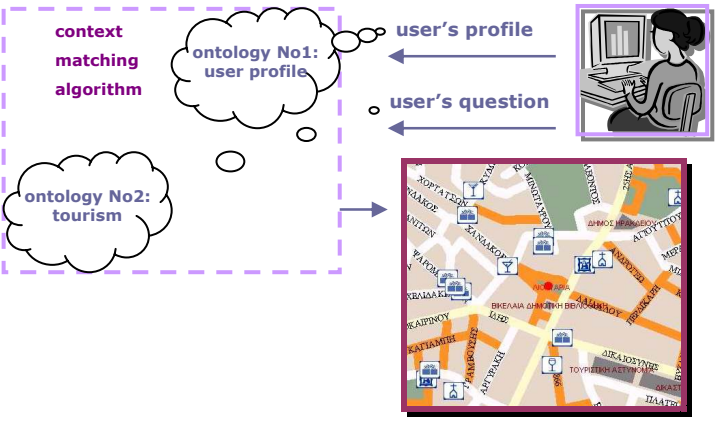


Fig. 1. The architecture of the system

3 User Profile Ontology and Interface

This section presents in detail the characteristics of the user profile ontology. The ontology was implemented using Protégé 3.0 [4] in OWL DL. The user profile ontology was created in order to facilitate the extraction of the user personal information, needs, and interests, under the context of personalization.

3.1 Personalizing the System

The key characteristic of the ontology is that it is comprised of two steps. The first step, that of the design, concerns agreeing upon the main concepts of the ontology along with their properties. We include in the ontology not only these concepts that characterize/describe a tourist but also concepts that account for the personal information of the user with respect to his/her trip making.

To be more specific we included in the ontology concepts such as: *age, gender, profession, leisure activities* and *interests* which sketch upon the personality of the user. Furthermore, concepts such as *kind of trip, time available, temporal period of the visit, accompanying persons, money to spend, transportation means* were added in order to reveal the characteristics of the user as a traveling agent. These concepts were further detailed by adding sub-concepts; for instance, for the concept *leisure activities* the sub-concepts *eating out, nightlife, shopping* and *sports* were created (The complete list of concepts of the user profile ontology is shown in fig. 2).

Based on each concept, a corresponding property was created. To make this clear to the reader, from the concept *interests*, the property *is interested in* can be created



Fig. 2. The concepts of the user profile ontology

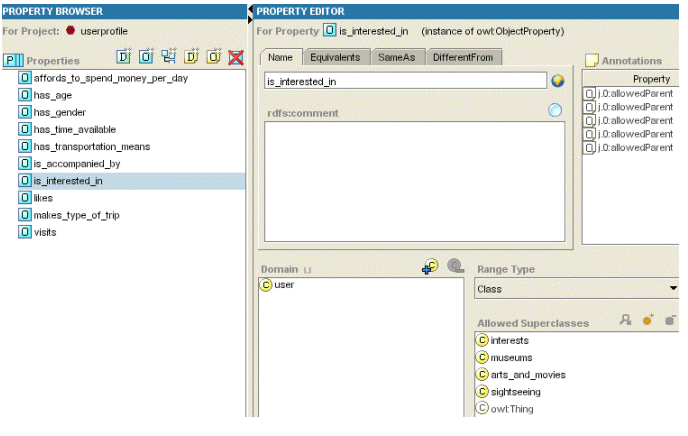


Fig. 3. The properties of the user profile ontology which are all assigned to the user/ tourist

which is assigned to the user and the values the property takes can all be found in the sub-concepts of the original concept, which is *interests* in this case. The properties of the ontology in this case play the role of posing questions to the user as a means to elicit to information from him/her. This "functionality" assists us in designing the user interface as it will be explained in the following section. For the previous example a question is: *what are you interested in?* And a possible answer from the user is: *I am interested in museums*. The complete list of attributes which are assigned to the user is shown in fig. 3.

3.2 Populating the User Profile Ontology

The second step is to populate the ontology with instances for the concept user. This is achieved, by providing an interface to the user so that he/she user can introduce personal information, interests and facts about the visit. The interface "resembles" a questionnaire and, as previously mentioned, it is web-based.

The procedure of collecting and recording the actual user profiles, in our case, is very much guided by the predefined user profile ontology. For example, when the user is asked to fill in his/her interests, he/she can only chose form a list of alternatives, given in the form of a drop-down menu, that correspond to the sub-concepts of the *interests* concept in the generic user profile ontology, presented earlier. This methodology has been previously presented in [6], for the creation of a web-based ontology editor. Fig. 4 presents a screenshot of the interface.



Fig 4. The first screenshot of the interface, where the user is asked to fill in personal data

The qualities of this methodology are two fold: it can elicit information on the user profile using the same terminology as the one of the generic user profile ontology, and also because the interface is structured based on the generic ontology; any information introduced therein can easily be recorded into the ontology as its instance (fig. 5). It can be easily understood that as more tourists use the system, the more the ontology gets populated. As a drawback, however, it should be pointed out that if the concepts of the generic ontology are modified, certain elements/pages of the interface should change to match the ontology. This downside really boils down to the adequacy and completeness of the original design of the ontology, which should minimize the risk of frequent changes.

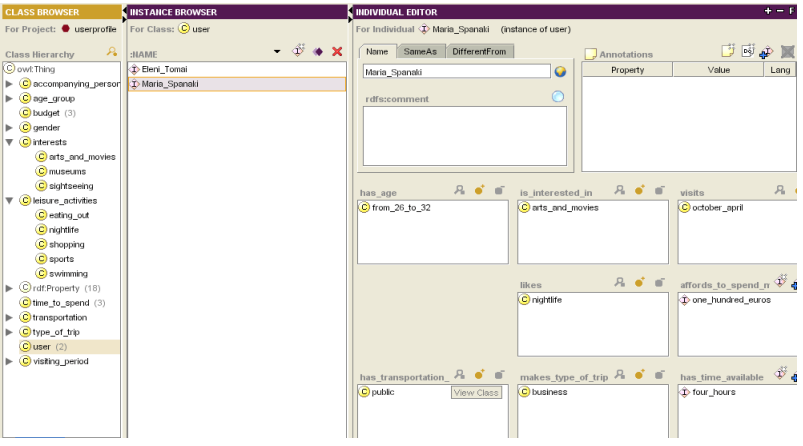


Fig. 5. An instance of a user in the user profile ontology

4 Ontology of Tourism

This section describes the second core component of the system, that of the tourism ontology. This encompasses concepts familiar to all tourists such as *sightseeing,*

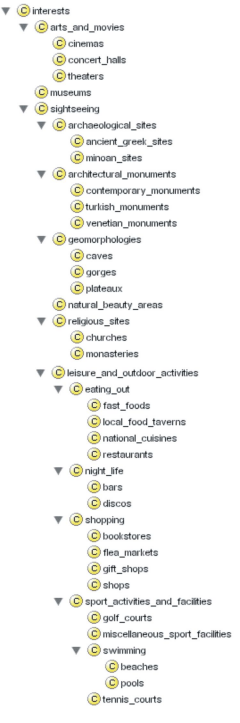


Fig. 6. The core concepts of the tourism ontology



Fig. 7. The additional concepts of the tourism ontology that help us assign properties to the concepts of fig. 6

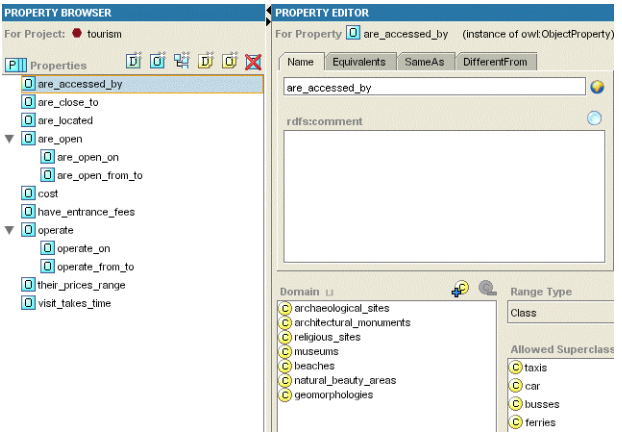


Fig. 8. The properties of the tourism ontology

shopping, leisure activities etc (fig. 6). The division of these fundamental concepts into sub-concepts, as those shown in fig. 6, was guided by the information of different websites and the classification of point of interest for tourists used therein such as the web pages of the city of Heraklion [3], the tourist guide for the municipality of Heraklion [7] and the agro-tourism site for Heraklion [1], available only in Greek.

On the other hand, the concepts of *location* and *time needed* have central role in the ontology. The former refers to either the location a point of interest has on a map, or its address, if that kind of information is available, while the latter refers to the time it takes for the tourist to get to the point of interest plus the average time to see the place and come back (the reference point for all users is taken to be the centre of the city of Heraklion).

Other concepts in the ontology concern additional information for the fundamental tourism concepts such as *accessibility, entrance fees, opening hours* and the like. Fig. 7 shows all the additional concepts included in the ontology. From those, we assign properties to the concepts of fig. 6. For instance, from the concepts *accessibility* we

create the property; *are accessed by* which involves the sub-concepts of *transportation,* and the concepts which are assigned this property are: *archaeological sites, museums, natural beauty areas* etc. properties help us set statements such as the following: *archaeological sites are accessed by busses*, or *beaches are accessed by taxis and ferries* etc. Fig. 8 demonstrates the list of the properties of the concepts included in the tourism ontology.

The tourism ontology is hidden form the user and it is populated with actual data as instances of the concepts included therein. This ontology was also implemented in Protégé in OWL DL.

5 Context Matching Algorithm

One of the basic points in the approach described above is the service that makes the semantic matching, e.g. a service discovery mechanism that can give results with high precision according to the user's queries. In location-based services the matching process involves the context, the user profile and the user history.

We propose a web service that can take into account existing parameters in LBS environment although the user is in a certain and static place when querying the system's database. To overcome problems emerging by the lack of user under way, we prompt him to give us information about the context by filling up a questionnaire. For example, questions like: *what is the time period you are visiting the X town?*, give the system an overview of the user profile.

As far as the location of the user is concerned our system works under the assumption that he/she is the city centre of Heraklion so all answers from the system concerning *distance* are measured from that point of reference.

When the user queries the system according to his/her interests and the time to spend, the semantic matching process starts by filtering out the services that do not match the service types asked by the user. The second step involves finding the correspondences between concepts and properties in the user profile and those in the tourism ontology. The use of common terminology in both ontologies speeds up the matching process and makes it easier.

On the third step, additional information provided by the user such as *visiting period* is taken into account and narrows even more the initial query, while in the last step, the matching result is classified in two modes; exact and approximate. Regardless the fact that the system finds a perfect matching or not, it is also able to give imperfect resulting sets as possible alternatives close to user's needs, the same approach has been proposed in [9].

In our approach, the user profile ontology is quite general in the sense that it has been designed in a way so that the corresponding interface which records all user information does not request from them detailed information for his/her interests or tastes. This design decision was taken on the basis that we wanted to provide to the users a list of alternative answers and let them take the final decision on how to spend their time. Another reason for keeping the user profile ontology quite generic is that our system does not tackle the issue of user history, therefore we needed to let the system give alternative answers on the assumption that it might not be the user's first time in Heraklion, consequently, a list of possible answers covers that aspect.

Crucial feature of the specific algorithm is the calculation of time. As already mentioned in the tourism ontology the concept of time (*time needed*) t_n encodes the time it takes for the tourist to get to a point of interest from the centre of Heraklion plus the average time to see the place and come back to the centre. While in the user profile ontology time (*time available*) t_a reflects the available time the tourist has to spend in Heraklion. Therefore if the *time needed* to visit a place of interest (t_{n1}) is less than *the time available* of the user (t_a) the system incorporates in the answer another point of interest that has time needed to visit it (t_{n2}). This process can go on as long as:

$$\sum_{i}^{n} t_n < t_a \text{ , where i to n are the points of interest} \quad (1)$$

For the equation to give more sophisticated results the concept of *proximity* should also be incorporated in the algorithm, which accounts for how close several points of interest are. Another concept which should be included in the algorithm when calculating time is that of *transportation means.* It is quite obvious that t_n changes whether the user has a car or he/she uses public transportation. From this discussion the calculation of time is quite critical for the conformity of the system's answer to the user needs.

6 Discussion and Further Work

Several approaches have been proposed with the intension of helping tourisms in exploring points of interest in a usually unknown geographic area. Most approaches use location-based services and event notification methods in mobile system. Our approach however, presents novelties such as the following:

1. The system is not a mobile service but a web service provided by a local authority, such as the Greek ministry of tourism, the municipality of Heraklion etc.
2. The information concerning tourist activities (data) are organized in an ontology not separate databases, so that the schema is quite generic, it can be expanded (further include more information).
3. Our main contribution is the interface where the user inputs his/her personal information so that the ontology of the user profile is elicited.
4. Moreover the terminology used in the interface is conformant to the terminology of the data ontology so that the matching algorithm is easier to implement and provide better results.

Further work concerns the inclusion of more parameters into the algorithm involving not only time available by the user, but also the amount of money he/she affords to spend during the visit. Moreover, several issues concerning the accuracy and availability of data that have not been yet treated should be investigated because they influence the structure and content of the ontologies.

References

1. Agro-tourism site for the municipality of Heraklion: http://www.in.gr/agro/Iraklio/nom1.htm
2. Galant V., Paprzycki M.: Information Personalization in an Internet Based Travel Support System. In: Abramowicz (ed.), Proceedings of the BIS'2002 Conference, Poland (2002) 191-202
3. Heraklion city's web pages: http://www.heraklion-city.gr/

4. Protégé Ontology Editor (2004) http://protege.stanford.edu/
5. ten Hagen K., Kramer R., Hermkes M., Schumann B., Mueller P.: Semantic Matching and Heuristic Search for a Dynamic Tour Guide, 12th International Conference on Information Technology and Travel & Tourism, Austria (2005)
6. Tomai E., Spanaki M.: From Ontology Design to Ontology Implementation: a Web Tool for Building Geographic Ontologies. In Toppen F., Painho M. (eds.): Proceedings of AGILE 2005, 8th Conference on Geographic Information Science, ISEGI-UNL, Portugal (2005) 281-290
7. Tourist guide for Heraklion: http://www.4crete.gr/creteguide/en_heraklion.htm
8. van Setten M., Pokraev S., Koolwaaij J.: Context-Aware Recommendations in the Mobile Tourist Application COMPASS. Lecture Notes in Computer Science, Vol. 3137 (2004) 235 - 244
9. Yu S., Al-Jadir L., Spaccapietra S.: Matching User's Semantics with Data Semantics in Location-Based Services, 1st Workshop on Semantics in mobile Environments, Cyprus (2005)
10. Yu S., Spaccapietra, S., Cullot, N., Aufaure M.: User Profiles in Location-based Services: Make Humans More Nomadic and Personalised, Proceedings of the International Conference on Databases and Applications, Austria (2004)

WOSE 2005 PC Co-chairs' Message

We are happy and proud to introduce the proceedings for this 2nd "Workshop on Ontologies, Semantics and E-learning". We use the term "ontologies" to refer to an as precise and formal as possible definition of the semantics of objects and their inter-relationships for a specific application domain.

It is interesting to note that the concept of ontologies has started to be applied in the context of learning technologies. Indeed, after some initial ontology related work in the late '80s and early '90s, mainly in the context of so-called "Intelligent Tutoring Systems", the focus over the last 10 years has been more on the development of learning objects, repositories and interoperability specifications, instead of on the formal representation and analysis of their meaning.

Important work remains to be done in several areas in order to enable e-learning to finally come of age. The analysis and application of the semantical layer in practical contexts of use is one of these areas, because it will help to tailor technologies to the specific requirements of learning communities.

We would like to warmly thank the members of our program committee, namely:

- Lora Aroyo, Eindhoven University of Technology
- Aldo de Moor, Vrije Universiteit Brussel, Belgium
- Erik Duval, Katholieke Universiteit Leuven, Belgium
- Robert Farrell, Next Generation Web Department, IBM Research
- Fabrizio Giorgini, Giunti Interactive Labs, Italy
- Ambjörn Naeve, Royal Institute of Technology, Stockholm
- Daniel Rehak, Learning System Architecture Lab, CMU
- Tyde Richards, Eduworks Corporation, Corvallis OR
- Peter Spyns, Vrije Universiteit Brussel, Belgium
- Frans Van Assche, European Schoolnet, Brussel
- Martin Wolpers, L3S Research Center, Hannover

In these proceedings, we also look forward to the somewhat farther future of e-learning. Learning environments become increasingly more important. So far, these are still mostly restricted to document and workflow management-like systems. However, one trend is that these environments provide ever more immersive, virtual reality-like experiences. We have therefore included a special track of papers on ontology mining and engineering and their use in virtual reality.

This session is chaired by Marie-Laure Reinberger (University of Antwerp) and Olga De Troyer (Vrije Universiteit Brussel) and organised by Peter Spyns (Vrije Universiteit Brussel). Because of its specific angle, this special session had its own program committee to which we are also greatly indebted:

- Galia Angelova, Bulgarian Academy of Sciences, Bulgaria
- Josep Blat, Universitat Pompeu Fabra, Barcelona, Spain

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1147–1148, 2005.
© Springer-Verlag Berlin Heidelberg 2005

- Paul Buitelaar, DFKI, Germany
- Marc Cavazza, University of Teesside, U.K.
- Frdric Kleinermann, Vrije Universiteit Brussel, Belgium
- Bernardo Magnini, ITC-irst, Italy
- Richard Sproat, University of Illinois, USA
- Victoria Uren, Open University, U.K.
- Esteban Zimanyi, Université Libre de Bruxelles, Belgium
- Pierre Zweigenbaum, A.P. - Hôpitaux de Paris, INSERM & INALCO, France

In total, we received 19 paper submissions. All of these papers were independently reviewed by at least three members of our program committees. In the end, we accepted 8 submissions. This implies an acceptance ratio of 42%. Most importantly, we believe that all the papers in this volume represent significant work that advances the state-of-the-art on the use of ontologies for learning.

We hope that you will find the papers that follow useful for your own understanding and practice of this new area in learning technologies research.

Best regards from the WOSE05 program chairs .

August 2005

Peter Spyns, Vrije Universiteit Brussel
Erik Duval, Katholieke Universiteit Leuven
Aldo de Moor, Vrije Universiteit Brussel
Lora Aroyo, Eindhoven University of Technology
(WOSE'05 Program Committee Co-Chairs)

Towards the Integration of Performance Support and e-Learning: Context-Aware Product Support Systems

Nikolaos Lagos, Rossitza M. Setchi, and Stefan S. Dimov

Manufacturing Engineering Center, Cardiff University,
Queen's Buildings, Cardiff CF24 3AS, UK
{LagosN, Setchi, Dimov}@Cardiff.ac.uk

Abstract. Traditionally performance support and e-Learning have been considered as two separate research fields. This paper integrates them by introducing the concept of context-aware product support systems, which utilizes the notions of context, information object and ontology. The context signifies whether learning or performing goals are pursued in different situations and defines the configuration of the domain knowledge accordingly. An information object is viewed as an enabler of a modular virtual documentation, advancing information reuse. The ontology formalizes the representation of the knowledge contained in the system, facilitates interoperability, and constitutes one of the main building blocks of context-aware product support systems. The prototype system developed illustrates the applicability of the approach.

1 Introduction

For many years, performance support and e-Learning have been considered as two separate research fields. Performance support is defined as a process that "aims to enhance user performance through a user interface and support environment…" [1], while e-Learning is "the use of new multimedia technologies and the Internet to improve the quality of learning by facilitating access to resources and services, as well as remote exchange and collaboration" [2]. The fundamental difference between these two areas is therefore inherent to their aims. Performance support targets results and efficiency when performing a specific task while e-Learning focuses on the acquisition of new skills and knowledge.

Studies show that there is a trend towards combining electronic performance support systems (EPSSs) and e-Learning by either transforming an EPSS into an e-Learning system by adding learning features or vice versa. Alstete [12] states that several characteristics of Web-based educational systems can be utilized to increase team performance like discussion boards and task lists. Alonso et al. [13] present an instructional model that enables learning not only based on teaching methods but by also realizing the assumption that "training should enable learners to apply the concepts learned in their workplace". Bareiss and Williams [14] utilize an EPSS to provide in-context help to students as they learn new concepts while Wild [15] supports novice teachers in the task of lesson planning. Dickover [16] identifies some learning characteristics of performance EPSSs such as explanations and lessons and Shaick et al. [17] facilitate learning by developing a task-driven student-centered

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1149 – 1158, 2005.
© Springer-Verlag Berlin Heidelberg 2005

EPSS. The aforementioned approaches propose a combination of EPSSs and e-Learning within the same context. However, they do not differentiate the tools and information provided in different contexts but rather target hybrid learning and performing.

Product support is defined as everything needed to support the continuous use of a product [3]. A product support system therefore aims to alleviate the lack of knowledge by the user in a particular subject or situation related to a given product. Traditionally, performance support techniques have been used in the development of product support systems. This study discusses a possible extension of their application area by proposing an integration of performance and learning goals within the context of a single product support system. The objective is to transform product support systems into a medium that not only offers just-in-time and personalized support but also enables users to improve their skill set by acquiring new knowledge. The following notions are utilized as enablers of the aforementioned objective.

1. *Context*. Modeling the product support system context of use and identifying contextual changes that delineates the way in which learning and performing can be linked.
2. *Information Object* (IO). IO as a documentation element advances reuse and composition of documents on-demand.
3. *Ontology*. Formal modeling based on ontologies enables structured reasoning to take place and interoperability issues to be considered.

The rest of the paper is organized as follows. Section 2 defines the context of use for a product support system. Section 3 introduces the virtual document model adopted in this study and its application for generating context-aware documents. Section 4 presents the ontological framework in which a contextual change is proposed. A prototype system is described in section 5. The final section contains conclusions and directions for further work.

2 Context of Use

2.1 Definition

The context of use has been defined as the piece of information that can be used to characterize the situation of an entity [4], the aggregation of factors that influence the user [5] or an application's environment [6], and an environment in which a task is carried out [7]. The context of use is hereby defined as a complete environment in which a user interacts with the system in order to alleviate his/her knowledge deficit in a particular situation related to a product. User is considered any person or group that directly interacts with the system. The context of use is represented with the following models.

1. *Activity Model* (or Purpose Model) (AM), which is a finite set $\{a_1, a_2, \ldots, a_n\}$ where a_i stands for a specific activity indicating the specific usage of the system. In this study two main abstract activities are discussed, 'perform' and 'learn'.
2. *User Model* (UM), which is a finite set $\{u_1, u_2, \ldots, u_k\}$ where u_i represents a user stereotype.

3. *Physical Model* (PM), which is a finite set $\{p_1, p_2, \ldots, p_m\}$ where each p_i stands for any hardware or software related property, such as an operating system or a graphics card memory.
4. *Environment Model* (EM), which is a finite set $\{e_1, e_2, \ldots, e_n\}$ where each element represents environmental conditions (e.g. location).

A context instantiation C_i, is denoted by the aggregation of the aforementioned models' instantiations and can be represented as $<a_i, u_i, p_i, e_i>$. In this study, the integration of the context of use with the domain knowledge is described in terms of the first two models (i.e. AM and UM).

2.2 Integration with Domain Knowledge

The domain knowledge of a product support system is represented with the task (TM) and the product (PM) models, along with their internal relations [8]. The task model includes all the tasks and subtasks that a user may require support (e.g. describe, install, and design) and the product model contains knowledge about the supported product(s). AM is used to develop particular configurations of the task model according to the variations in the activity selection. For example, as illustrated in Fig. 1, if a user selects the 'Learn' activity, the task model includes the 'Describe', 'Promote', 'Assess', 'Design', and 'Plan' tasks and their corresponding subtasks. On the other hand, if a user selects the 'Perform' activity, the task model is (re)configured and contains the 'Design', 'Plan', 'Operate', 'Inspect', and 'Install' tasks, as well as their corresponding subtasks. The activities (context) are related to the tasks (domain) with the spatial relation 'IsRealisedWith', which defines the current configuration of the task model. For instance, using McCarthy's formalization [9], the relation 'IsRealisedWith' between the 'Learn' activity and the 'Describe' task, is equivalent to the following.

$$c0: \quad ist\ (c_{activity}(learn), \text{“describe is a task”}) \tag{1}$$

(1) asserts that in the "Learn" activity context "Describe" is a task. c0 is considered as the outer context of a product support system. As illustrated in Fig. 1, some tasks may be related to different contexts at different times (e.g. 'Design'). In such cases, the generated documents should reflect the contextual difference by including the elements that are related to the current context, as presented in the next section.

3 Context-Aware Virtual Document

IO is the smallest constituent of a virtual document. It is defined as "a data structure that represents an identifiable and meaningful instance of information in a specific presentation form" [10]. IO can therefore be a picture that illustrates a part of a product or a textual description. In this study IOs are characterized according to their form, behavior, type, expressiveness, and theme.

1. *Form* indicates whether IO is a text, image, animation, video, audio, etc.
2. IO can have two forms of *behavior*, namely static or dynamic. Static behavior indicates that IO remains the same under all circumstances (e.g. the definition of a

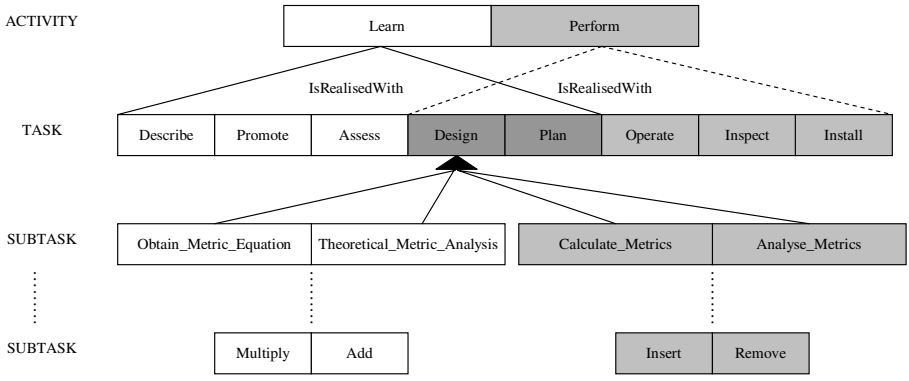


Fig. 1. Relation between the context (activity model) and domain knowledge (task model)

clutch). Dynamic behavior means that IO changes at run-time, according to the attribute of the real world object it describes (e.g. the radius of a clutch).

3. *Type* defines whether IO is an element of explanation, description, definition, etc.
4. *Expressiveness* is a qualitative measure that describes the way in which an IO manifests content (e.g. detailed vs. concise).
5. The *theme* identifies the concept that IO refers to (e.g. clutch).

The notion of Information Object Cluster (IOC) is introduced, as a means of organizing IOs. IOC is defined as a 2-tuple IOC:=({IO}, S_{IOC}) where {IO} is a set of IOs sharing a common property that are arranged in a structure S_{IOC}. A structure defines the way in which they are presented within the same page, as well as the relevant links. Theme, expressiveness, and type are used to identify all the IOs that belong to the same IOC. S_{IOC} conforms to presentation rules (e.g. a textual description should always appear before the corresponding image).

The Virtual Document (D) is generated by the aggregation of IOCs and is defined as a 2-tuple D:=({IOC}, S_D) where {IOC} is a set of IOCs sharing a common property that logically structured (S_D) in order to compose a document (D). The IOCs are selected and organized according to their theme, context, and type, as follows.

1. *Theme configuration*. The combination of different themes that is required within the same document. For example, a document about the installation of a clutch, has two major themes, which are the task 'Install' and the product 'Clutch'. Since the installation is further divided into smaller subtasks and/or steps the IOCs corresponding to them are also required. Each step refers to the initial product (i.e. clutch) but also to other subcomponents (e.g. bolt), associated with the IOCs. The structure of the document follows that of the product and task. For example, a document for clutch design includes design subtasks, such as "devise metrics", and for each of them other subtasks (i.e. constituent steps), such as "calculate metric". In this example, the metrics are related to the product "clutch", such as "torque", so

the corresponding textual description "devise metrics" is substituted by "devise torque". The links are created according to keywords that describe other product and task concepts (e.g. transmission).

2. *Context selection*. The context within which the document is created is defined by UM and AM. The expressiveness of the selected IOs depends on the user category (e.g. for novice user detailed IOs are presented). According to the activity choice, different type categories may be selected (e.g. quizzes are included in learning context). Links are created between different contexts for each IOC, enabling the user to change context at run-time.

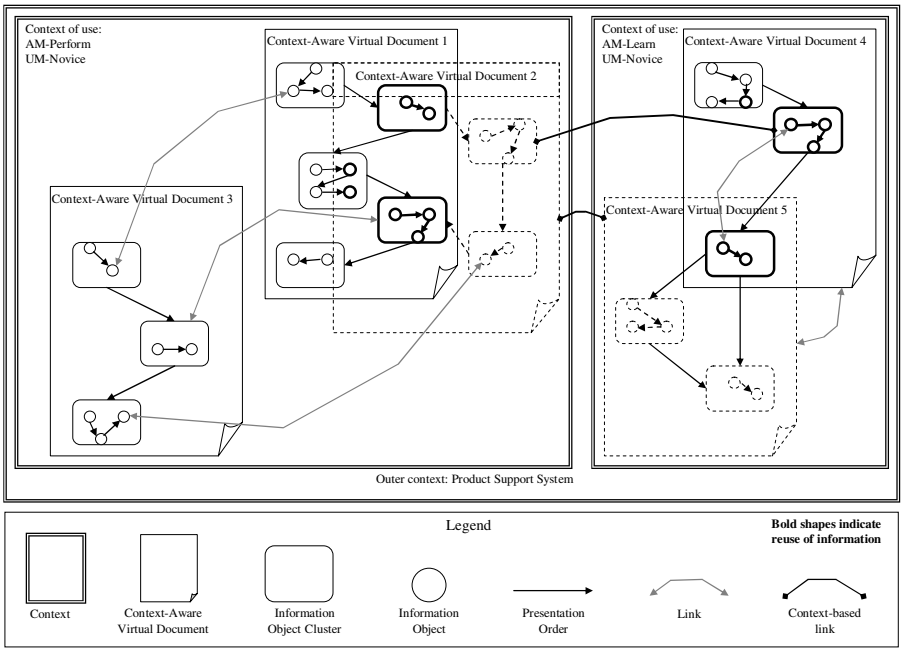


Fig. 2. Model of context-aware virtual documents and their relations

3. *Type ordering*. The type category defines the order in which the IOCs are presented in the document. For example, a title should be at the top, before a description, while a definition should be followed by an example.

The aforementioned segmentation of a virtual document advances the reuse of information (i.e. reuse of IOs and IOCs) for generating different documents. The model of context-aware virtual documents and their relations (Fig. 2) enables the ontology-based formalization of the knowledge contained in a context-aware product support system.

4 Ontology-Based Representation of Knowledge in Context-Aware Product Support Systems

A context-aware product support system contains knowledge about the domain (task and product models), the context (e.g. activity model, user model) and the documentation elements. As a result, the development of such a system is highly interdisciplinary. In order to advance interoperability between these different areas and product support, an ontology that formalizes the aforementioned knowledge has been developed. A part of the ontology is illustrated in Fig. 3.

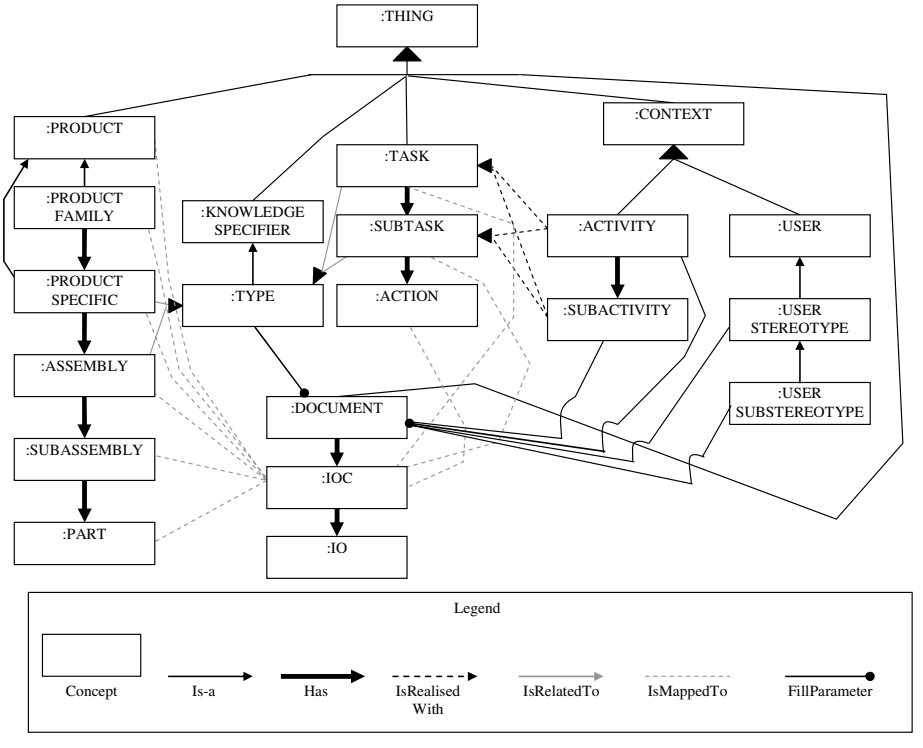


Fig.3. Part of the ontology for context-aware product support systems

The ontology can be described according to the product, task, context, and document models it includes.

- The product model represents the structure of the product. All its concepts are mapped to IOCs as explained in the rest of the section. Concepts ":PRODUCT SPECIFIC" and ":ASSEMBLY" are linked to the concept ":TYPE". This is a specialization of :"KNOWLEDGE SPECIFIER", which abstracts all concepts that represent domain significant properties. For example, the type of an assembly, i.e. whether it is considered as complex or not [11], affects the generation of the document (this will not be further discussed here as it is not within the focus of this paper).

- The task model contains the tasks, subtasks, and actions that are supported, where action is the most elementary step of a task. All are mapped respectively to IOCs and are related to ":TYPE". Furthermore, the task model is configured according to its relation to the activity model (see section 2.2).
- The context model includes the activity and user models (more models can be included as explained in section 2.1). Both models are related to the ":DOCUMENT" concept with the relation "FillParameter", which denotes that the characteristics of the user and the activity are passed as adaptation parameters to the document. Furthermore, the activity model also defines the variations of the task model.
- The document model embodies the structure of a virtual document, as introduced in this paper. The "IsMappedTo" relation signifies the content by mapping each IOC to a different concept. Moreover, each slot (or each attribute) of a concept, is mapped to at least one IO. Slots that are dynamic (i.e. can have different values) are linked to dynamic IOs, while the rest are connected to static ones. The document that is generated includes all the IOCs, and therefore concepts, that the query requires. The context of use is passed to the document as a set of parameters that define both content and presentation aspects (e.g. novice user requires detailed descriptions).

The ontology provides a means to represent the elements from which required knowledge is constructed in a machine processable way and is the basis of a context-aware product support system. It is considered as the building block that will enable structured reasoning to take place.

5 Case Study

A prototype system that realizes the proposed approach has been developed. The ontology has been employed for the creation of a knowledge base, which is built with the Protégé environment and currently includes more than 200 concepts. Other enabling tools include Apache's Tomcat servlet container, JBoss application server, FreeCBR case-based reasoning tool, and Java.

The scenario discussed is that a user requests information from the context-aware product support system about the clutch design procedure. It is assumed that the user is inexperienced and wants to design a clutch (i.e. perform a task). The system responds by generating the top left document (screenshot) in Fig. 4. The virtual document includes IOCs that correspond to the task "design", its subtasks, and the product "clutch". IOs are selected also in terms of the context (i.e. "novice"-"performing"), which is passed as a set of parameters to the document generation process. In this case, the task process is configured as follows: Determine goal → Identify constraints → Make preliminary calculations of metrics → Analyze layout of metric attributes → Revise and make final calculations of metrics. The goal, constraints metrics, metric attributes, and calculations are specific to the supported product, which in this case is the "clutch". Example metrics are the torque and moment of inertia of the clutch. In order to support the realization of the current task, performance tools are provided,

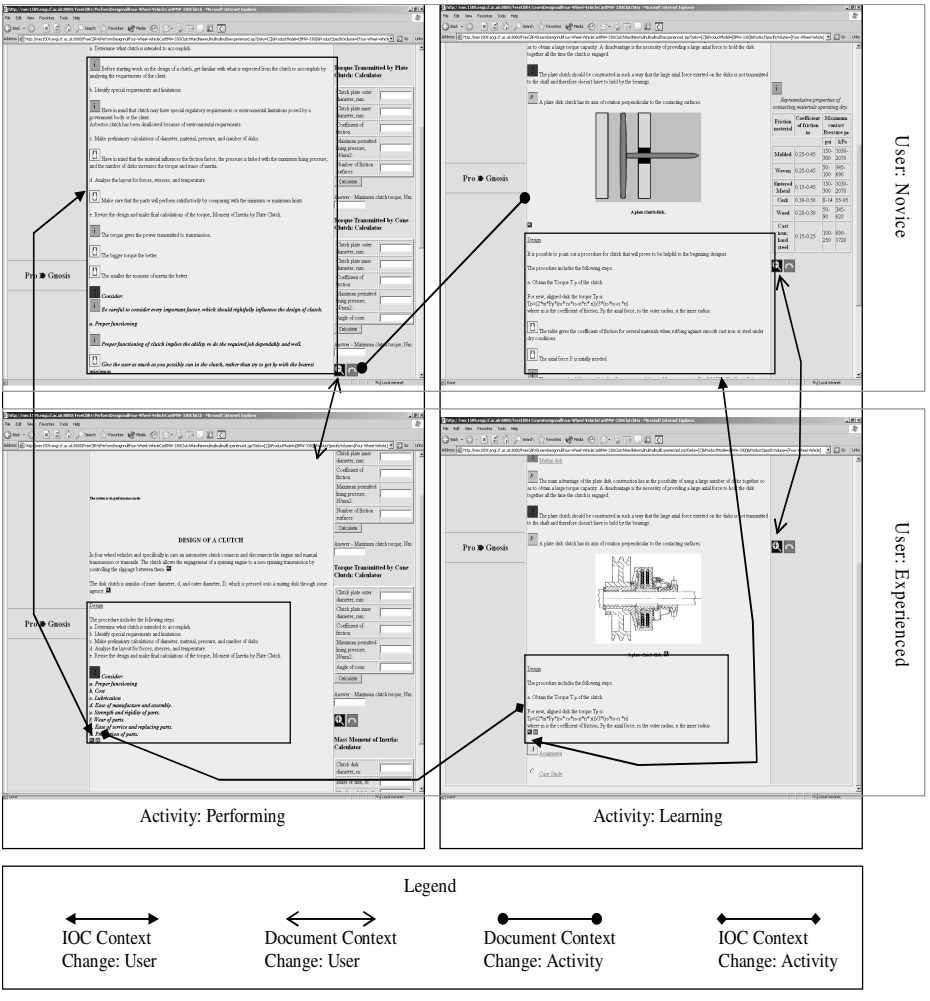


Fig.4. Context-based change between different documents and Information Object Clusters (IOCs)

which in this case include a calculator for the torque and one for the moment of inertia (right frame of the top left screenshot). The presented IOs are 'detailed', animations and video are preferred over still images, and include all clarification types such as 'information', 'rule_of_thumb', and 'explanation'.

Part of the user's training is to learn about the theoretical background of the clutch design metrics. This means that the context should change from 'performing' to 'learning'. "Context-change" buttons are developed within the generation procedure of the document that link IOCs and the whole document to other contexts. If the user clicks the "activity-change" button, the task procedure is reconfigured in the following one: Obtain the main metric equation → Obtain other metrics equations that exist within the initial equation → Present theoretical aspects of these equations → Solve

related assignments and case studies (or Evaluation). The new document is illustrated in the top right screenshot in Fig. 4. The "performance tools" are replaced by "learning tools" such as summarization tables. The presented IOs are 'detailed', include animations and videos, and all clarification types such as 'descriptions' and 'examples' (note that the clarification types change, e.g. "rule-of-thumb" is not used in learning).

The user can also change the category level at run-time through the "user-change" button. This removes all clarification objects, changes the expressiveness to 'general', and includes more "technically-oriented" textual descriptions and still images, as shown in the bottom right screenshot in Fig. 4. The utilization of the "context-change" buttons enables the user to change context at run-time and the provision of support becomes highly adaptive and interactive.

6 Conclusions and Future Work

The research presented in this paper integrates performance support and e-Learning within the field of context-aware product support systems. The integration is achieved by configuring the task model according to the context of use and by defining context-dependent parameters as factors of the documentation. A model of a context-aware virtual document is introduced that enhances the reuse of information by segmenting the document into smaller constituents, namely Information Objects and Information Object Clusters, which identify the way in which content, structure, and context aspects are applied throughout the document generation process. The delineation of context and documentation facilitates the formalization of the knowledge contained within a context-aware product support system and its ontology-based representation. The developed ontology advances the interoperability between documentation and other knowledge intensive fields, such as product and context modeling, and forms the basic building block towards the development of context-aware product support systems.

Future work includes the investigation of the ontology fragmentation into smaller easier manageable ontologies with the utilization of intelligent software agents, which will allow distributed computation to be performed. The long term objective is to enable performance and educational learning to become possible within a single highly distributed and continuously evolving system, which will allow seamless collaboration among dispersed groups of people.

Acknowledgements. The research described in this paper was supported by Cardiff University and performed within the I*PROMS Network of Excellence sponsored by FP6 of the European Community.

References

1. Bezanson, W.R.: Performance support: online, integrated documentation and training. Proc. 13th Int. Con. Systems Documentation: Emerging from Chaos: Solutions for the Growing Complexity of our Jobs, SIGDOC' 95. ACM Press, New York (1996) 1-10

2. Commission of the European Communities (COM2001). The eLearning Action Plan: Designing Tomorrow's Education. Communication for the Commission to the Council and the European Parliament. Brussels (2001) 1-19
3. Pham, D.T., Dimov, S.S., Setchi, R.M. "Intelligent Product Manuals" Proc. IMechE, J. Sys. Con. Eng. 213 (1 I), (1999) 65-76
4. Abowd, G.D., Dey, A.K., Brown P.J., Davies, N., Smith, M., Steggles, P.: Towards a Better Understanding of Context and Context-Awareness. In: Gellersen H.W. (eds.): Handheld and Ubiquitous Computing. Lecture Notes in Computer Science, Vol. 1707. Springer-Verlag, Berlin Heidelberg New York (1999) 304-307
5. Schmidt, A., Beigl, M., Gellersen, H.W.: There is more to Context than Location. Workshop on Interactive Applications of Mobile Computing IMC'98 (1999)
6. Chen, G., Kotz, D.: A Survey of Context-Aware Mobile Computing Research. Dartmouth Computer Science Technical Report TR2000-381 (2000)
7. Souchon, N., Limbourg, Q., Vanderdonckt, J.: Task Modelling in Multiple Contexts of Use. In: Forbrig, P., Limbourg, Q., Urban, B., Vanderdonckt, J. (eds.): Interactive Systems. Design, Specification, and Verification. Lecture Notes in Computer Science, Vol. 2545. Springer-Verlag, Berlin Heidelberg New York (2002) 59-73
8. Setchi, R.M., Lagos, N., Dimov, S.S.: Semantic Modeling of Product Support Knowledge. I*PROMS Virtual Conference. Springer-Verlag Berlin Heidelberg New York (2005)
9. McCarthy, J.: Notes on Formalizing Context. Proc. 23rd Int. Joint Con. AI IJCAI'93, Chambery, France (1993) 555-560
10. Pham, D.T., Setchi, R.M. "Authoring Environment for Documentation Elements" Proc. IMechE, J. Sys. Con. Eng. 215 (1 B), (2001) 877-882
11. Benton, W.C., Srivastava, R. "Product structure complexity and inventory storage capacity on the performance of a multi-level manufacturing system" Int. J. Prod. Res. 31 (1993) 2531-2545
12. Alstete, J.W. "Alternative Uses of Electronic Learning Systems for Enhancing Team Performance" Int. J. Team Per. Man. 7 (3/4), (2001) 48-52
13. Alonso F., Lopez G., Manrique D., Vines J.M. "An Instructional Model for Web-Based e-Learning Education with a Blended Learning Process Approach" British J. Ed. Tech. 36 (2), (2005) 217-235
14. Bareiss, R., Williams, S.M.: ASK Jasper: Performance Support for Students. In: Conf. on Human Fact. in Comp. Sys. ACM Press, New York NY (1996) 183-184
15. Wild, M. "Designing and Evaluating an Educational Performance Support System" British J. Ed. Tech. 31 (1), (2000) 5-20
16. Dickover, N.T. "The Job is the Learning Environment: Performance-Centered Learning to Support Knowledge Worker Performance" J. Int. Inst. Dev. 14 (3), (2002) 1-8
17. van Scaick, P., Pearson, R., Barker, P. "Designing Electronic Performance Support Systems to Facilitate Learning" Inn. Ed. and Teach. Int. 39 (4), (2002) 289-306

Taking Advantage of LOM Semantics for Supporting Lesson Authoring

Olivier Motelet* and Nelson A. Baloian

Universidad de Chile, Santiago, Chile
Universidad Diego Portales, Santiago, Chile
{omotelet, nbaloian}@dcc.uchile.cl

Abstract. Learning Object Metadata (LOM) is an interoperable standard aimed to foster the reuse of learning material for authoring lessons. Nevertheless, few work was done on taking advantage of LOM-semantics to facilitate retrieval of learning material. This article suggests an original approach which uses the structure of a lesson in order to automatically generate LOM-semantic-based queries for retrieving learning material for that lesson whereas the user continues to formulate easy-to-write queries without semantic specifications. This proposal consists of a four-component framework attempting to consider the main issues of semantic-based retrieval of documents.

1 Introduction

One of the main motivations behind Learning Objects and Learning Object Repositories is to facilitate their reuse by as many people as possible. In order to make this possible, the characteristics of the learning objects should be exposed, so that other people could locate and retrieve them. A very critic issue in this process is how to describe an object and how to search for it in order to find those who really would match the needs of a potentially user. The metadata describing a learning object is a fundamental characteristic enabling this process. In order to make the finding of a suitable learning object more accurate, the description of a learning object should not only consider the physical characteristics of the document, like the one proposed by the DublinCore Metadata Intitiative[1] but it should also be pedagogically relevant. The Learning Object Metadata (LOM) standard includes such data. Consequently, Learning Object Repositories (LOR) typically use this metadata for the storage and retrieval of learning objects. However, following this standard means that authors or people classifying learning objects should assign values to almost 60 metadata attributes in order to fully describe the material according to the IEEE LTSC LOM specification[2]. Also users trying to retrieve the learning material may have to deal with this problem. Such a fastidious task is not compatible

* This work has been financed by Chile-Corea IT Cooperation Center.
[1] http://www.dublincore.org/
[2] http://ltsc.ieee.org/wg12/

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1159–1168, 2005.
© Springer-Verlag Berlin Heidelberg 2005

with making learning material sharing a customary activity for regular teachers. Several researchers have already described this problem and propose the automatic generation of metadata as a way to solve it[1,2,3]. Basically, metadata generation systems are intended to improve the performance of metadata exploitation systems[4]. Similarly, metadata exploitation system should influence metadata generation system specifications. However, the topic of exploiting the metadata of learning objects is still in its beginnings. The typical way of making use of the metadata for retrieving relevant learning material is making a query ala Google on all the attributes independently of their nature. More advanced exploitation systems called recommender systems make use of the experience and opinion of other people having already used this material (see [5] for an example). Baloian et al.[6] use LOM and user/system modeling as a base of a collaborative recommender system for learning material. Duval and Hodgins[1] suggest a collaborative filtering system based on rating and pattern recognition. These systems benefit from the semantics of LOM, i.e. the semantic structure of the data, to rate the didactic material and facilitate its retrieval. This article presents an approach that benefits from LOM semantics for retrieving learning objects to fit in a certain learning context. This procedure is aimed to support an instructor during the authoring of an entire course syllabus based on learning material retrieved from different repositories without having to provide explicitly all the metadata values for querying the repository. Moreover, this approach may also help to automatically generate metadata for a learning object which exists inside a coherent course syllabus. Since this method is based on the existence of a graph that structures and relates the learning material to support the process, it is complementary to the use of recommender systems. In order to introduce our work, learning object retrieval based on LOM semantics is discussed. Then, learning object graphs are brought in and their dependency with LOM semantic is studied. Next, integration and processing of LOM-semantic-based retrieval is presented. Finally, a framework of a system implementing our approach is drawn.

2 LOM-Semantic-Based Retrieval of Learning Object

Google and other indexing engines typically provide interfaces for simple queries with a semantic based on logical operators. In such systems, these basic queries (*BQueries*) target complex indexes generated by document content analysis procedures. Learning Object Repositories generally offer interfaces for processing such simple queries. In these settings, BQueries concern all the elements of the objects' metadata set independently of their nature, making a string matching without using any semantic similarity of the terms. Although such a retrieval process is simple for the end-user, it does not benefit from one of the main advantages of metadata over indexes: their semantic classification. LOM exploitation systems should use this characteristic to overcome the limits of string-based indexing engines. In the currently existing LORs, users have to complete forms with all the fields of the learning objects metadata set in order to make a query considering which takes in account the metadata semantics . Indeed, query languages

enabling semantic precision (for example XPath, XQuery or RDFQL) are too complex to be integrated at user-level. Form-based queries for retrieving learning objects is a time consuming and tedious task. Studies[7] show that authors of learning material do not properly generate complete and correct metadata. In the same way, we do not expect that users are willing to properly generate metadata for searching this kind of material. Processing semantic based queries involves many well-known problems characterized by the Artificial Intelligence[8]. In particular, a system for processing LOM-Semantic-based Queries (*LSQueries*) should be able to find relations between the vocabulary used in the query expression and the vocabulary used in the learning object repositories. If no relevant matching can be found, LSQueries should be approximated in order to effectively retrieve the desired material. Approximated outcomes could be reached for example by a process in which query restrictions are relaxed according to predefined or customizable strategies.

Some work dealing with the automatic production of queries enabling semantic-based retrieval of learning objects has been motivated by the difficulty of doing it 'by hand'. Typically, this kind of systems falls in the category of recommender systems based on complex recognition pattern methods or user profile analysis[6,1]. Other systems assist users to generate LSQueries. Pinkwart et al.[9] present a system generating LSQueries based on the potential similarities existing between learning material. This method is particularly aimed to support collaborative learning. Learning Management Systems (LMS) could also help users to generate LSQueries by providing information like the educational context, the expected learning time and the used language[1]. Our approach takes advantage of the structure in which a learning object might be embedded in order to enable semantic-based retrieval of learning objects. For example, a syllabus of a certain learning unit might be represented with a graph, in which nodes contain the learning material and edges the relations between them.

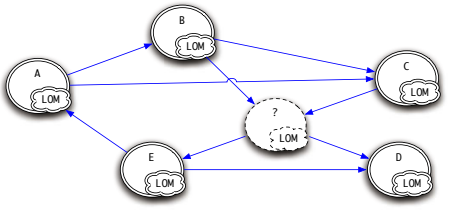


Fig. 1. Graph of learning objects during authoring process. One element is not yet referring a concrete material, but it is already characterized inside the graph.

As illustrated in figure 1, a lesson syllabus graph could refer to both, already existing material and material that has still to be provided or retrieved. Our proposal is to take advantage of the semantics of the graph, i.e. the nature of the edges relating the nodes, in order to identify some characteristics of the missing material. This process should enable the retrieval of educational resources suiting to the context defined by the lesson syllabus. In order to introduce this approach, the next part explores the influence of graph semantics on the semantics of LOM.

3 Influence of Graph Semantics on LOM Semantics

From the beginning of the development of intelligent application for learning, authors have proposed the structuring of learning material in graphs. In [10], McCalla presents a number of self-adapting tutoring systems for supporting individual learners and he considers the graph as a key structure for the learning unit syllabus in order to achieve flexibility. Fischer[11] uses two different graphs to define a syllabus. First, a graph of concepts is built by means of a set of semantic relations. Second, a graph of material is defined based on a set of rhetorical relations. Using learning material metadata, the system generates semi-automatically the sequencing of the learning material. Baloian et al.[12,13] use a graph structure for representing the syllabus of a learning unit. Such graphs are called Didactic Networks. Didactic Networks enable the generation of several versions of the same lecture according to different teaching styles and learning requirements. This functionality is based on the analysis of a predefined set of rhetoric relations between the didactic activities. Similarly, Farrell et al.[14] suggests to enable the dynamic assembly of learning objects using graph of concepts and LOM semantics. In particular, the LOM semantics are used to identify the rhetorical relations linking the learning objects.

Independently of the type of graph used for structuring the learning material, an obvious fact is that the relations between two educational resources depend on their type and content. By definition, LOM values should reflect these characteristics. Consequently, in a learning object graph, the relations between two elements depend on the values of their metadata. Reversely, the values of the metadata of two learning objects may somehow be aligned with the relationships existing between these elements.

Consider the two learning objects $L1$ and $L2$ of figure 2. $L1$ theorizes $L2$ and $L2$ concretizes $L1$. Some *similarities* between the values of their metadata can be observed. For example, values for keywords, educational context, and classification are quite alike in both educational resources. This fact is not just a coincidence: we could derive the values of some metadata of one learning object from others by

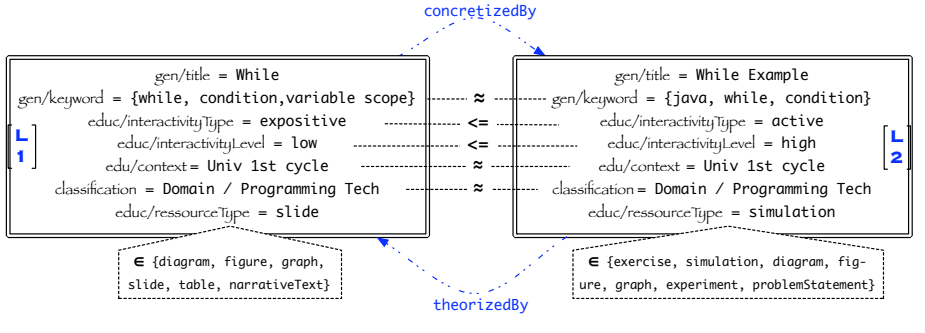


Fig. 2. Two learning objects $L1$ and $L2$ linked with rhetoric relations. These relations imply mutual influences between the LOM document values.

considering the relations between them. For instance, the nature of the relations between $L1$ and $L2$ imposes some *restrictions* on the material nature. Since $L2$ concretizes $L1$, $L2$ will deal with an exercise, a simulation, a diagram, an experiment, or a problem statement, whereas $L1$ will deal with a diagram, a figure, a graph, some slides, a table, or a narrative text. For the same reason, type and level of $L1$'s interactivity with the students will be certainly lower or equal than the ones of $L2$. Perhaps these assumptions may not have been valid for all potential users, so each community should define their own rules according to their needs. The important fact is that such rules provide relevant information for retrieving the learning material which is missing in a lesson graph. In particular, some rules may generate *restrictions* on the values of certain metadata. These restrictions could be used to formulate the queries to be sent to learning object repositories. In addition to that, other rules identify *similarities* between certain metadata. These similarities may serve to rank the query results. In the next part, a framework for semantic-based retrieval of learning objects during lesson authoring is presented.

4 Semantic-Based Retrieval during Lesson Authoring

Document retrieval systems are basically composed of two main components feeding mutually each other, one dealing with query processing and the other with result processing. Query processing provides results and result processing may define new queries. In addition to them, a framework for semantic-based retrieval of learning objects during authoring of lesson syllabus should consider not only a component reflecting the authoring process, but also a component responsible for the generation of semantic-based queries and ranking information.

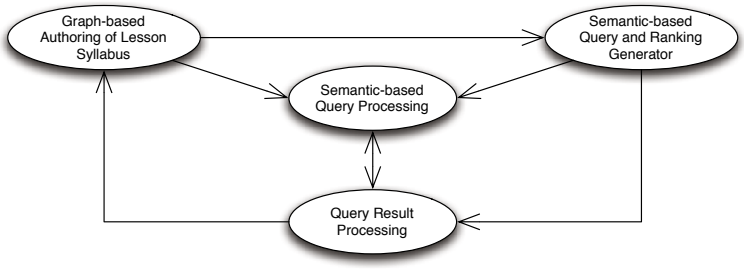


Fig. 3. Framework for semantic-based retrieval of learning objects during authoring of lesson syllabus.

4.1 Graph-Based Authoring of Lesson Syllabus

The lesson graph component is responsible for supporting the authoring of the lesson syllabus. In the current implementation, this component is based on a

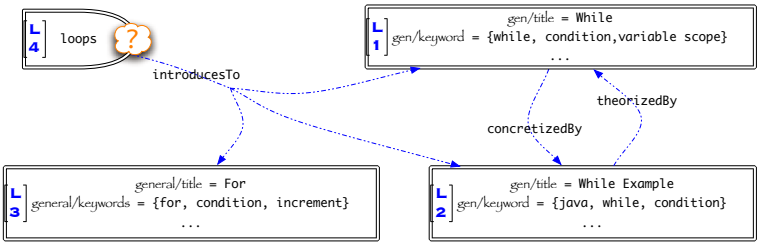


Fig. 4. L1, L2, and L3 are learning objects with LOM description. The instructor is looking for a learning object L4 in order to introduce L1, L2, and L3.

previous work: LessonMapper[13,15], a Java application for authoring lesson graphs. This platform is extended to assume the process illustrated by figure 4. In this example, the author of the syllabus of a lesson about programming languages is looking for some learning material in order to introduce the concept of loops. First, s/he creates a new node $L4$ characterized with the key-sentence `loops`. Then, s/he specifies that $L4$ introduces $L1$, $L2$, and $L3$ by creating links of type `introducesTo`. On one hand, the key-sentence is used to formulate a BQuery (see section 2), which is provided to the query processing module. On the other hand, the graph semantics is processed by the generator component in order to create a LSQuery reflecting the lesson context.

4.2 Generator of Semantic-Based Query and Ranking Information

The generator component is intended to provide semantic-based queries to the query processing component and also ranking information to the result processing module. Various implementations of this component may be developed. For instance, this module may process pattern analysis or user profile matching, or simply recover some relevant information from LMS. This work focuses on the proposal of the previous section: taking advantage of the influences of the graph semantics on LOM semantics. In this approach, processing the semantics of a lesson graph with a set of rules generates queries and ranking information. This section described the lanaguage used for specifying these rules and the framework applied to diffuse them recursively throughout the graph.

Generation Rule Specification Language. As argued before, generation rules should be well suited to the teaching/learning habits of the potential users. Therefore, they should not be hard-coded in the system, but defined with a easy-to-use language permitting local customizations of the system. Because a generic language like OWL would be too complex to suit for this, we choose to define our own Domain Specific Language (DSL). Our implementation includes the mathematical operators: *max*, *min*, *union*, *intersection*, *sum*, *product*, *subtraction*, and *division*. In the example of figure 4, since $L4$ introduces to $L1$, $L2$, and $L3$, the `keyword` metadata of $L4$ may have some similarities with the `keyword`

metadata of $L1$, $L2$, and $L3$. In order to postulate such a statement, the user should define the following rule:

```
<similarity attribute="general/keyword" relation="introducesTo">
          UNION(v)
</similarity>
```

in which v stands for the value set of `keyword` metadata of all the educational resources related with `introducesTo`
In order to specify restrictions on metadata values, our language also provides a set of comparison operators: $=$, $!=$, $<$, $<=$, $>$, $>=$, *contains*, and *containedIn*. For instance, since $L4$ introduces to $L1$, $L2$, and $L3$, the semantic density of $L4$ should be inferior or equal to the minimum semantic density of $L1$, $L2$, and $L3$. Such a restriction is generalized with the following rule:

```
<restriction attribute="general/semanticDensity" relation="introducesTo">
          '<=' MIN(v)
</restriction>
```

In order to compute this kind of rules, comparable elements should provide an order value. In the present implementation, RDF vocabulary includes such a value as shown in the following example:

```
<lom_edu:InteractivityLevel rdf:ID="MediumInteractivity" order="15"/>
<lom_edu:InteractivityLevel rdf:ID="HighInteractivity" order="20"/>
```

Nevertheless, since most learning object repositories are not able to process restrictions based on vocabulary comparison, such properties is then expressed in terms of sets of values. For instance, a restriction of type '$\leq$ `mediumDifficulty`' is transformed in '$\in \{$`veryEasy`, `easy`, `mediumDifficulty`$\}$'.

Generation Rule Diffusion. The difficulty to properly generate metadata for educational resources imposes to consider potential incompleteness of LOM values in the lesson graph. In order to deal with this situation, we suggest to take advantage of the graph structure and propagate restrictions and similarities through the whole graph. In our implementation, this propagation process is done recursively based on the framework introduced in a previous work [16]. In this framework, restrictions and similarities are not only based on the metadata of other educational resources, but also on the set of restrictions and similarities generated for these resources. Basically, this model introduces propagation and composition principles for restrictions and similarities. This feature enables the recursive processing of the rules and it limits the side-effect of metadata incompleteness.

4.3 Semantic-Based Query Processing

This component is responsible for defining a query Q combining the contributions of the authoring component and the generator component. Afterward, Q is distributed to a set of learning object repositories.

Query Formulation. On one hand, the authoring component generates a BQuery, i.e. a query not considering any semantic restriction. On the other hand, the generator component provides semantic restrictions formulated as LSQueries. First, Q is defined as a conjunction: $Q = \texttt{BQuery} \bigwedge \texttt{LSQuery1} \bigwedge ... \bigwedge \texttt{LSQuery}n$. Later, Q may be relaxed (for example in a disjunction) by the query result processing component. In our prototype, the queries are formulated with Xquery. However, the choice of a query language should first depend on the compatibility with the targeted repositories.

Query Distribution. Query distribution deals basically with the communication with the learning object repositories. In our present implementation, distribution is limited to a local repository. Nevertheless, distribution should definitely be considered in order to reach enough sources for making a retrieval system interesting. An interface like Simple Query Interface (SQI)[17] may support this process. If we expect teaching/learning communities to use specific local vocabulary for sharing the educational resources[3], the terminology used in the query may differ from the one used in the learning object repository. However, such usage involves complex vocabulary distribution and interoperability issues[8].

4.4 Query Result Processing

The query result processing component deals with the answers of the consulted repositories. First, it is responsible for the presentation of the results. Then, according to the subjective analysis of the user, the first query may be relaxed and/or some didactic material may be reused.

Result Organization. The generator component is responsible for supporting the ranking of the results returned by the learning object repositories. In our implementation, ranking information is based on the similarity set produced by processing the generation rules, in which educational material matching more similarities has a better rank than other material. Such a service may be also implemented with collaborative filtering techniques. Moreover, information visualization techniques may efficiently support the user in browsing the query results[18].

Query Relaxation. In case that results are too few because of the restrictions imposed by the generator, the lesson author could reformulate some part of the generated query by relaxing restrictions imposed on some attributes. For example, restrictions imposed on the `general`, `lifeCycle`, `technical`, and `classification` categories of LOM may be relaxed to enlarge the search to educational resources matching with a certain pedagogical context but not limited to a specific discipline or format. The learning objects resulting from this relaxation process may offer interesting hints for defining methods supporting the particular educational goal of the authored lesson. Further work should be done on LOM semantics in order to offer a set of pedagogically-sounded relaxation strategies.

Learning Object Reuse. Once the syllabus designer has selected one or more learning objects to be reused, a recontextualisation phase is required. This process deals with the adaptation of the retrieved material to a specific use context. Format, language, style and copyrights issues have to be managed, but these topics remain far out of the scope of this article.

5 Conclusion

This article presents an original approach for enabling LOM-semantic based retrieval of learning objects during the lesson authoring process. Our proposal differs from existing semantic-based retrieval systems because it is based on the analysis of the semantics of the graph structuring the whole lesson in which the retrieved learning material is intended to be included. For that reason, it can be used along with other approaches based on user profile, pattern analysis, or material similarity. For the same reason, our system focuses on a specific context: lesson-syllabus structuring based on graph.

Lesson graphs are not specific to this work but explored by several researches in the community. Their main advantages are flexibility during course presentation and semantic-based sequencing of the lesson. We attempt to aggregate another advantage to lesson graphs: the semantic-based retrieval of learning objects. Our approach enables automatic generation of LOM-semantic-based queries, whereas the user continues to formulate easy-to-write queries without semantic restrictions. Such system is based on generation rules exploiting the influences of graph semantics on LOM semantics. The same approach is also used to rank the query results according to the context of the authored lesson. The model can be adapted to specific didactic behaviors since rules are defined with a simple domain-specific language. Moreover, a recursive diffusion framework limits the impact of a potential incompleteness of the learning material metadata.

This learning object retrieval system is part of a four-component framework capable of integrating other methods for generating queries and ranking data. This framework includes learning object retrieval as a legitimate component of the lesson authoring process. Query and result processing are considered with the perspective of semantic-based retrieval of educational resources. Complex issues like vocabulary distribution and learning object re-contextualization remain open. Nevertheless, interesting perspectives are also emerging like the possibility to define pedagogically-sound retrieval strategies.

References

1. Duval, E., Hodgins: Metadata matters. In: International Conference on metadata and Dublin Core specifications DC 2004, Shangai. (2004)
2. Simon, B., Dolog, P., Miklós, Z., Olmedilla, D., Sintek, M.: Conceptualising smart spaces for learning. Journal of Interactive Media in Education- Special Issue on the Educational Semantic Web. **5** (2004)

3. Downes, S.: Ressource profiles. Journal of Interactive Media in Education, Special Issue on the Educational Semantic Web. **5** (2004)
4. Brasher, A., Andrew, P.M.: Human-generated learning object metadata. R. Meersman et al. (Eds.) OTM Workshops 2004 **LNCS** (2004) 723–730
5. Rafaeli, S., Dan-Gur, Y., Barak, M.: Social recommender systems: Recommendations in support of e-learning. J. of Dist. Educ. Tech. **3** (2005) 29–45
6. Baloian, N.A., Galdames, P., Collazos, C., Guerrero, L.: A model for a collaborative recommender system for multimedia learning material. In LNCS, ed.: Groupware Design, Implementation and Use: CRIGW. Volume 3198. (2004) 281–288
7. Friesen, N.: International lom survey report. Technical report, ISO/IEC JTC1/SC36 sub-committee (2004)
8. Stuckenschmidt, H.: Query processing on the semantic web. Kunstliche Intelligenz: Special Issue on the Semantic Web **3/03** (2003) 22–26
9. Pinkwart, N., Jansen, M., Oelinger, M., Korchounova, L., Hoppe, U.: Partial generation of contextualized metadata in a collaborative modeling environment. In: 2nd International Workshop on Applications of Semantic Web Technologies for E-Learning AH 2004, Eindhoven, Netherlands. (2004)
10. McCalla, G.: The search for adaptability, flexibility, and individualization: Approaches to curriculum in intelligent tutoring systems. In: Foundations and Frontiers of Adaptive Learning Environments. Springer (1992) 91–122
11. Fischer, S.: Course and exercise sequencing using metadata in adaptive hypermedia learning systems. J. Educ. Resour. Comput. **1** (2001) 5
12. Baloian, N.A., Hoppe, H.U., Pino, J.A.: A teaching/learning approach to cscl. In: 33rd Hawaii International Conference on System Sciences HICSS. (2000)
13. Baloian, N.A., Pino, J.A., Motelet, O.: Collaborative authoring, use, and reuse of learning material in a computer-integrated classroom. In: CRIWG. (2003) 199–207
14. Farrell, R., Liburd, S.D., Thomas, J.C.: Dynamic assembly of learning objects. In: World-Wide Web International Conference WWW 2004, New York. (2004)
15. Motelet, O., Baloian, N.A.: Introducing learning management systems standards in classroom. In Kinshuk, Looi, C.K., Sutinen, E., Sampson, D.G., Aedo, I., Uden, L., Kähkönen, E., eds.: International Conference on Advanced Learning Technologies ICALT 2004, Sueden, IEEE Computer Society (2004)
16. Motelet, O.: Relation-based heuristic diffusion framework for lom generation. In: Artificial Inteligence in Education AIED 2005 - Young Researcher Track. (2005)
17. Simon, B., Massart, D., Duval, E.: Simple query interface specification. In: CEN/ISSS Workshop on Learning Technologies. (2004)
18. Klerkx, J., Duval, E., Meire, M.: Using information visualization for accessing learning object repositories. In: 8th International Conference on Information Visualization. (2004)

Repurposing Learning Object Components

Katrien Verbert[1], Jelena Jovanović[2], Dragan Gašević[3], and Erik Duval[1]

[1] Dept. Computerwetenschappen, Katholieke Universiteit Leuven,
Celestijnenlaan 200A, B-3001 Leuven, Belgium
{Katrien.Verbert, Erik.Duval}@cs.kuleuven.be
[2] FON-School of Business Administration, University of Belgrade,
POB 52, Jove Ilica 154, Belgrade, Serbia and Montenegro
jeljov@gmail.com
[3] School of Interactive Arts and Technology, Simon Fraser University Surrey,
2400 Central City, 10153 King George Hwy., Surrey, BC V3T 2W1, Canada
dgasevic@sfu.ca

Abstract. This paper presents an ontology-based framework for repurposing learning object components. Unlike the usual practice where learning object components are assembled manually, the proposed framework enables on-the-fly access and repurposing of learning object components. The framework supports two processes: the decomposition of learning objects into their components as well as the automatic assembly of these components in real-world applications. For now, the framework supports slide presentations. As an application, we will present in this paper the integration of this functionality in MS PowerPoint.

1 Introduction

Learning objects are often stored in a final presentation form. Such a static representation is neither suitable for flexible content reuse, nor adaptable to the needs of a learner. In many cases, specific parts are assembled manually by copy and paste actions. However, it is possible to reuse learning objects in a much more sophisticated way if their components can be accessed on-the-fly. This requires a more innovative and flexible underlying model for learning object components [2].

In earlier work, we developed an ontology (ALOCoM) for learning objects that is a framework for learning objects and their components [10]. The ontology defines learning object component types, as well as relationships between these components. As such, the ontology enables structuring of composite learning objects and is a solid basis for the proposed dynamic approach. In this paper, we describe a framework that transforms existing learning objects from their tool specific formats (MS Office, OpenOffice.org) into a representation compliant with the ALOCoM ontology. In this transformation process, the framework disaggregates learning objects and provides direct access to their components, enabling their reuse in dynamic compositions of new learning objects.

For now, the framework supports slide presentations as they are one of most common used learning object types [8]. Often a teacher wants to repurpose a

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1169–1178, 2005.
© Springer-Verlag Berlin Heidelberg 2005

slide or an image, a reference, a definition or just a text fragment of a particular slide. In our approach, we decompose the presentation into these individual components and store them into the learning object repository. As such, these components are accessible on-the-fly. The disaggregation of a slide presentation proceeds in three steps. In the first step, the presentation is parsed and decomposed into clear segments, namely its slides and each slide is further decomposed into its title, paragraphs, lists, list items, images, diagrams and tables. In the second step, text patterns are applied to categorize these segments into more meaningful components like definitions, examples, references, introductions and summaries. Finally, components are described by metadata using the Automatic Metadata Generation framework [1], improving the findability of relevant components.

Having in mind the widespread use of MS PowerPoint [8], we have created a MS PowerPoint add-in that allows searching for components in the learning object repository from within the MS PowerPoint application. When a teacher is creating a new slide presentation, (s)he can search for definitions, slides, examples, references and images (s)he wants to repurpose. Available components are displayed on the Clipboard and can directly be added to the slide presentation.

In the next section, we briefly outline the ontology that formalizes learning object types and their components. In section 3, we present the ALOCoM framework and section 4 illustrates the transformation of MS PowerPoint slide presentations. Section 5 provides more details about the classification of components and section 6 elaborates on annotating components. In section 7, we will demonstrate the integration of the ALOCoM framework into MS PowerPoint. Related work is presented in section 8 and conclusions and remarks on future work conclude this paper.

2 The ALOCoM Ontology

In earlier work, we developed the ALOCoM ontology as a generic Abstract Learning Object Content Model (ALOCoM - see Figure 1) for learning objects and their components [10]. The ontology distinguishes between content fragments (CFs), content objects (COs) and learning objects (LOs). CFs are learning content elements in their most basic form, like text, audio and video. These elements are uncombined with other elements. COs aggregate CFs and add navigation. Navigation elements enable structuring of CFs in a CO. Besides CFs, COs can also include other COs. Finally, LOs aggregate COs around a learning objective.

We defined content types for each of these components. We introduced CF types such as an image, text, and audio and video sequences. For defining CO types, we investigated existing Information Architectures, like the Information Block Architecture developed by Dr. Horn [3] and the IBM Darwin Information Typing Architecture (DITA) [9]. These architectures define information types (e.g. concept, principle, task) and their building blocks (e.g. example, definition, analogy). As a starting point, we defined CO types and their structure using DITA concepts, since DITA is a recent architecture with rich documentation

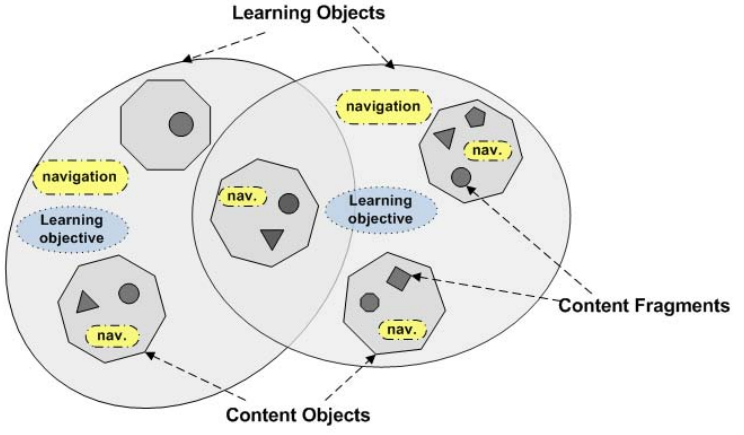


Fig. 1. Abstract Learning Object Content Model

and online support [9]. Besides CF and CO types, the ontology identifies LO types. For now, only a slide presentation LO type is defined. Finally, the ontology defines the relationships between the LO components. Aggregation and navigation relations are specified. Aggregation relationships between components are represented in the form of a "hasPart" and its inverse "isPartOf" ontology properties. Navigational relationships are specified as a list that defines the order of components in a CO or LO. For more information about the ontology, see [4].

3 An Ontology-Based Framework for Component Repurposing

Our main focus is on the development of tools for disaggregating learning objects into their components (i.e. disaggregators) as well as for repurposing learning object components in real-world applications (i.e. aggregators). For now, we developed a framework that provides both functionalities for slide presentations. Since the most popular tools for slide presentation authoring are MS PowerPoint and OpenOffice.org [8], the proposed framework focuses currently on slides presentations authored using these tools. The framework decomposes MS PowerPoint and OpenOffice.org slide presentations and assembles components into new MS PowerPoint and OpenOffice.org slide presentations on-the-fly. The disaggregation and re-aggregation processes are illustrated in Figure 2. In the disaggregation process, a slide presentation is parsed and disaggregated into clear segments (slides, paragraphs, lists, list items, images, diagrams and tables). In the second step, these segments are categorized into more meaningful content objects like definitions, examples, references, summaries, overviews and introductions. We use text patterns to classify these content objects. In the last step, content objects are annotated using the AMG framework [1]. These transformation steps will be further explained in the next three sections.

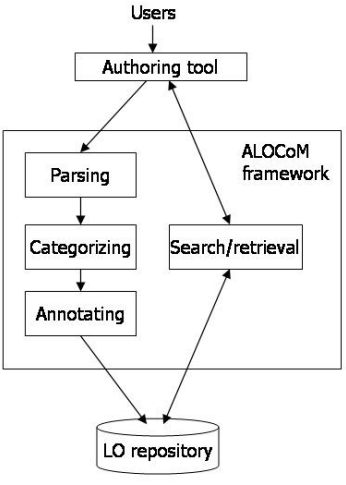


Fig. 2. The ALOCoM framework

The opposite (aggregation) process searches for components in the learning object repository and adds them to a slide presentation. Since authors prefer to use authoring environments they are familiar with, this functionality has to be integrated in present authoring tools. We have currently developed an add-in for MS PowerPoint. The same functionality will be provided for OpenOffice.org. Section 7 illustrates the use of the PowerPoint add-in, while the next one gives implementation details of the procedure for disaggregating slide presentations created in MS PowerPoint.

4 Parsing MS PowerPoint Slide Presentations

In the first transformation step, MS PowerPoint slide presentations are disaggregated into structured components. A slide presentation is decomposed into its slides and each slide into its title, paragraphs, lists, list items, images, diagrams and tables. We use the Microsoft PowerPoint .Net API for this transformation.

Microsoft Office Presentation objects are arranged in a hierarchical order, as shown in Figure 3. The two main classes at the top of the hierarchy are the Application and Presentation classes. An Application object provides a wrapper around the entire application. Each Presentation object represents a single Presentation document. Each of these objects has many methods and properties that allow manipulating and interacting with it. A Presentation has a Slides property that returns a collection of all the Slide objects in the presentation. A single slide is retrieved by specifying its name, index number or slide ID number. Each slide has a Shapes property that returns a Shapes collection representing

all the elements that have been placed or inserted on the specified slide, slide master, or range of slides. This collection can contain drawings, shapes, OLE objects, pictures, text objects, titles, headers, footers, slide numbers, and date and time objects appearing on a slide, or on the slide image on a notes page. Text objects contain presentation related information, enabling us to infer the structure of the text. For instance, if the Bullet property of a paragraph is set to visible, we transform the text fragment into a list. More generally, all content, structure and presentation related information is retrieved and transformed into an explicitly structured format.

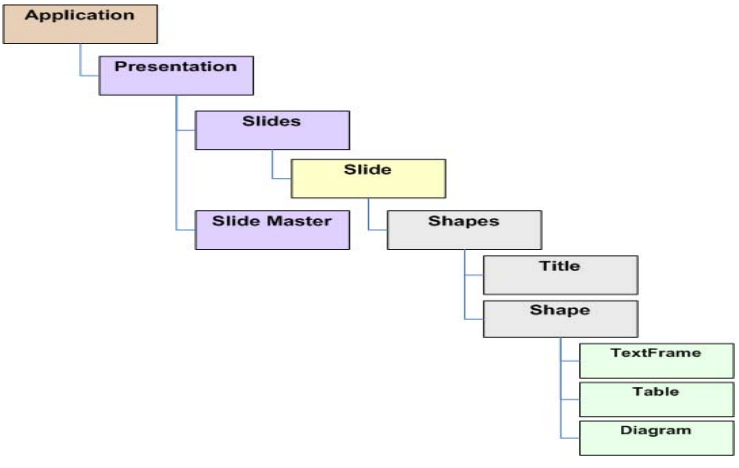


Fig. 3. PowerPoint Object Model

Figure 4 shows an excerpt of a MS PowerPoint slide presentation after being transformed into an ALOCoM compliant format.

```
<SlidePresentation>
<slide>
<title>Overview</title>
<slidebody> <!–...–> </slidebody>
</slide>
</SlidePresentation>
```

Fig. 4. A slide presentation in the ALOCoM XML format

5 Categorization of Content

The next step involves the categorization of the segments of the previous step (paragraphs, lists, list items) into more meaningful content object types, like

definitions, examples, references, overviews, introductions and summaries. Due to the lack of strict compliant rules for creating content, this is not a trivial task. However, for the categorization of some of these content object types (definitions, concepts) research has been done and solutions exist. For instance, a system called Definder uses rule-based techniques to extract definitions from medical articles [7]. Definder uses cue-phrases (e.g. *is the term for*, *is defined as*, *is called*) and text markers (e.g. $- -$, ()) in conjunction with a finite state grammar to extract definitions. Based on this system and research of Bing [6], we identified patterns that are suitable for recognizing definitions, examples and references in presentation slides.

5.1 Definition Extraction

The following patterns are applied to identify definitions of concepts:

1. {is|are} [*adverb*] {called | known as | defined as} {*concept*}
2. {*concept*} {refer(s) to | satisfy(ies)} ...
3. {*concept*} {is|are} [*determiner*] ...
4. {*concept*} {is|are} [*adverb*] {being used to | used to | referred to | employed to | defined as | formalized as | described as | concerned with | called} ...
5. {What is} [*determiner*] {*concept*}?
6. {*concept*} {- | : } {*definition*}
7. {definition} [of] {*concept*} {- | : } ...

Legend:
{} - compulsory field
[] - optional field
adverb - e.g., usually, normally, generally,...
determiner - e.g., the, one, a, an, ...
definition - definition of a concept.

For example, using pattern number five from the presented patterns list, content of a slide with title *"What is an ontology?"* is categorized as a definition of the ontology concept. Similarly, a list item containing the text *"an ontology is a specification of a shared conceptualization"*, will be classified as being a definition according to the third pattern. Although some authors use braces (e.g. () <> []) to wrap definitions, they are not used to detect definitions in our work. Braces are also used to wrap examples, illustrations and descriptions, so they will not help us in distinguishing between these components.

5.2 Example Extraction

The following patterns are applied to identify examples of concepts:

1. {example, instance, case, illustration, sample, specimen} [of] {*concept*}
2. {for instance | e.g. | for example | as an example} [,] [*determiner*] {*concept*} ...
3. {*concept*} {illustrates | demonstrates | shows | exemplifies} ...
4. {*concept*} {is|are} [*adverb*] {illustrated by | demonstrated by | shown by} ...
5. {Example} {- | : } {*example*}

Legend (new items):
example representing an example of a concept.

5.3 Reference Extraction

There are more strict guidelines for references, what makes identifying them easier. For instance, references are often preceded by the sequence "[identifier]", where identifier is a number or character sequence. An other standard uses the sequence "Name (Year)" to start the reference. This results in the following two identification patterns for references:

1. {[}{*identifier*}{]} {*reference*}
2. {*Name*} {(}{*Year*}{)} {*reference*}

Legend:
identifier - number or character sequence, e.g., 1, 2, Nam01 ...
reference - literature reference.

5.4 Summary, Overview and Introduction Extraction

Since we are currently working with slide presentations, we can easily classify introductions, summaries and overviews by looking at the title of slides. Slides are classified as summaries if their title is *"conclusion", "summary", "future work"* or a combination of these values. Introduction and overview slides are in most cases entitled respectively *"introduction"* and *"overview"*.

6 Annotating Components

The last step of the transformation consists of annotating the learning object components. We are using the Automatic Metadata Generation (AMG) framework to automatically describe each component (http://ariadne.cs.kuleuven.ac.be/amg). The idea behind the framework is to combine metadata, generated by different sources into one metadata instance [1]. The first source is the learning object itself; the second is the context in which the learning object is used. Metadata derived from the object itself is obtained by content analysis, such as keyword extraction and language classification. The contexts typically are learning (content) management systems (like Blackboard) or author institution information. A learning object context provides additional information about the learning object that is used to define the metadata. In our case, we developed an extension of the

framework that combines metadata by an inheritance mechanism. The metadata describing a component is also defined by the parents of this component. For instance, each slide inherits the author, language, etc. from the slide presentation it belongs to. Other metadata fields like for instance the title are overwritten. Also the main concept of a slide is in many cases more specific than the main concept of the slide presentation as a whole. Furthermore, dependency relations between learning objects and their components are described as relation metadata within a relation element. Through additional attributes the relation element specification allows distinguishing different relations between parent and child components ("isPartOf", "hasPart") and between components ("ordering"). As such, we defined an extension of the AMG framework that deals with an inheritance mechanism and relationships between components.

7 An Application: The ALOCoM Framework Integrated in MS PowerPoint

We have created an add-in for MS PowerPoint (Figure 5), which allows authors to repurpose components stored in the ALOCoM learning object repository without leaving their authoring environment (in this case MS PowerPoint). The add-in enables authors to search the repository for learning object components they wish to repurpose in the slide presentation they are working on. An author can specify the type of component he/she is interested in (e.g. reference, definition, example, slide, image), as well as keywords or other metadata fields that best

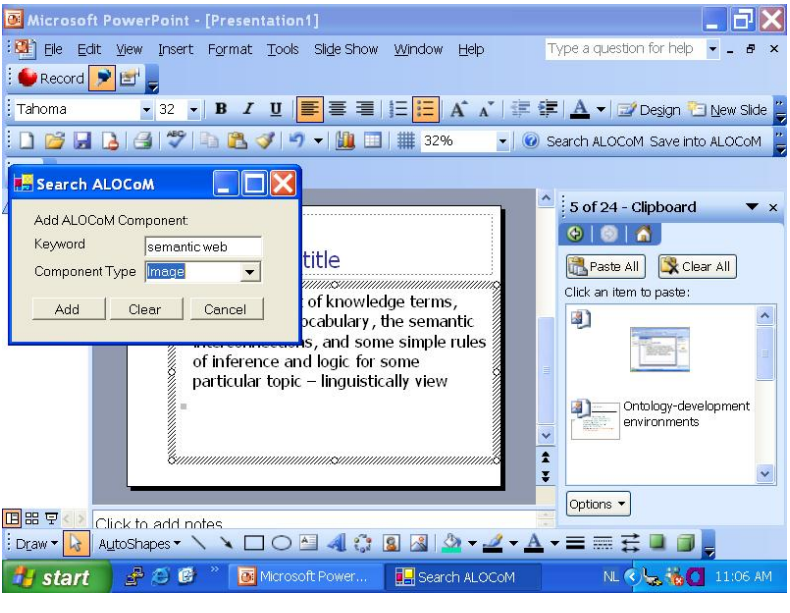


Fig. 5. MS PowerPoint - ALOCoM add-in

describe the component. All components that satisfy the specified search criteria are added to the Clipboard and the author can easily incorporate them into the current slide presentation. Another functionality provided by the developed add-in is related to adding new content to the learning object repository. Each author can add his or her slide presentation to the repository by clicking the "Save into ALOCoM" button. When this button is clicked, the PowerPoint presentation is disaggregated and stored into the ALOCoM LOR.

A typical use case goes as follows: Suppose an author is creating a slide presentation on ontologies. He/she wants to start with a definition, followed by three examples. The author enters "ontology" as keyword and selects "definition" and "example" as types of components that he/she is interested in. The system then searches the LOR and retrieves all components of the selected types dealing with the selected topic. The components that are found are added to the Clipboard. The author chooses the most relevant components from the set prepared for him/her. Furthermore, the author wants to include a reference to a book (s)he wants to recommend and an image of the book. Again the author searches the LOR and selects the component he/she wants to repurpose from the set of retrieved components. The author enhances the slide presentation with an additional example of an ontology and the presentation is ready for in-class use. Finally, the author clicks the "Save into ALOCoM" button. The slide presentation is disaggregated and also all (new) components of this presentation are available in the repository.

8 Related Work

The TRIAL-SOLUTION project is developing tools to create and deliver personalized teaching materials that are composed from a library of existing documents on mathematics at undergraduate level [5]. Analogously to the ALOCoM work, the TRIAL-SOLUTION project defines an ontology for learning objects that includes mathematical categories like definition, theorem, proof, or example. The focus of the project is on document (de-)composition and exchange of learning objects for reuse. The TRIAL-SOLUTION System contains a splitter that decomposes document source files into a hierarchy of slices. For these decomposition, the presentation style of a particular author is taken into account. Also, it takes care of counters and key phrases assigned by the author. As such, the methodology for decomposing learning objects is more accurate but less scalable than the methodology presented in this paper.

9 Conclusions and Future Work

In this paper, we have shown how we can improve present learning object authoring tools (e.g. MS PowerPoint) by integrating functionalities that allow on-the-fly repurposing of learning object components. The developed prototype validates this approach for slide presentations. In the next steps, the framework will be

extended to support a broader range of learning objects. Furthermore, the efficiency and effectiveness of this approach for learning object repurposing will be evaluated. This work will then result in a general framework for reusable learning objects, that allows not only automatic repurposing of learning objects, but also their components and that will enable the dynamic generation of learning objects, adapted to the needs of learners.

Acknowledgements

We gratefully acknowledge the financial support of the Katholieke Universiteit Leuven research council through the BALO project and of the European Commission through the ProLearn Network of excellence, which has facilitated the collaboration between the Computer Science Department of the K.U. Leuven, the GOOD OLD AI group at the University of Belgrade and Simon Fraser University Surrey (SFU). The research of SFU is also funded by LORNET, a NSERC Research Network.

References

1. K. Cardinaels, M. Meire, and E. Duval. Automating Metadata Generation: the Simple Indexing Interface. In *Proceedings of the fourteenth international conference on World Wide Web*. ACM, 2005.
2. E. Duval and W. Hodgins. A LOM research agenda. In *Proceedings of the twelfth international conference on World Wide Web*, pages 1–9. ACM Press, 2003.
3. R. E. Horn. Structured writing as a paradigm. In *Instructional Development: the State of the Art. Englewood Cliffs, N.J.*, 1998.
4. J. Jovanovic, D. Gasevic, K. Verbert, and E. Duval. Ontology of learning object content structure. In *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, 2005.
5. W. Lenski and E. Wette-Roch. The TRIAL-SOLUTION Approach to Document Re-use Principles and Realization. In *Electronic Media in Mathematics*, 2001.
6. Bing Liu, Chee Wee Chin, and Hwee Tou Ng. Mining Topic-Specific Concepts and Definitions on the Web. In *Proceedings of the twelfth international conference on World Wide Web*. ACM Press, 2003.
7. S. Muresan and J.K. Klavans. A Method for Automatically Building and Evaluating Dictionary Resources. In *Proceedings of the Language Resources and Evaluation Conference*, 2002.
8. J. Najjar, S. Ternier, and E. Duval. The Actual Use of Metadata in ARIADNE: An Empirical Analysis. In *Proceedings of the 3rd Annual ARIADNE Conference*, pages 1–6, 2002.
9. M. Priestley. DITA XML: a reuse by reference architecture for technical documentation. In *Proceedings of the 19th annual international conference on Computer documentation*, pages 152–156. ACM, ACM Press, 2001.
10. K. Verbert and E. Duval. Towards a global architecture for learning objects: a comparative analysis of learning object content models. In *Proceedings of the ED-MEDIA 2004 World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pages 202–209. AACE, AACE, 2004.

Interoperable E-Learning Ontologies Using Model Correspondences

Susanne Busse

Computation and Information Structures (CIS),
Technische Universität Berlin, Germany
sbusse@cs.tu-berlin.de

Abstract. Despite the fact that ontologies are a good idea for interoperation, existing e-Learning systems use different ontologies for describing their resources. Consequently, exchanging resources between systems as well as searching appropriate ones from several sources is still a problem.

We propose the concept of *correspondences* in addition to metadata standards. Correspondences specify relationships between ontologies that can be used to bridge their heterogeneity. We show how we use them for building evolvable federated information systems on e-Learning resources in the World Wide Web. Beside this integration scenario, we also describe other interoperation scenarios where correspondences can be useful.

1 Introduction

Our information society requires the continuous learning of each of us to gain necessary qualifications for our everyday life. In the last decade many activities have started to support this learning process: Learning content as well as tools for its development and management have been developed.

But despite some standardisation efforts in the e-Learning community (f.e. the LOM metadata standard [1] or the SCORM Reference Model[1]), the integration and interoperation of systems and its content is still a problem because

- there exists *no common ontology* that can be used to describe and to search e-Learning objects. Instead, each collection of learning objects is described using a local ontology or a local adaptation of a standard.
- the systems are *autonomous* and in consequence *heterogeneous*: The descriptions are based on different data models, differ in their data structures and semantics. The systems provide different interfaces and can change independently from each other and from systems using them.

Based on Web technologies the activities on the *Semantic Web*[2] address the challenge of searching and integrating data with different semantics using concepts from databases and information retrieval ([2,3]). The transfer of concepts of federated database systems results in the definition of *mediator-based information systems (MBIS)*.

[1] Sharable Content Object Reference Model of the Advanced Distributed Learning Initiative, http://www.adlnet.org/

[2] see http://www.semantic-web.org and http://www.w3.org/2001/sw/

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1179–1189, 2005.
© Springer-Verlag Berlin Heidelberg 2005

An MBIS offers a homogeneous, virtual, and read-only access mechanism to a dynamically changing collection of heterogeneous, autonomous, and distributed data sources ([4,5]). The user does not have to know how to combine data sources and how to integrate the results – the system encapsulates the heterogeneity and provides a search interface with a query language and a schema as a common ontology. As mediator ontologies we find top level standards like the Dublin Core[3] or domain-specific standards like the Learning Object Metadata standard in the e-Learning domain.

The main software components of an MBIS are *wrappers*, which encapsulate sources and solve technical and data model heterogeneity, and *mediators*, which integrate data from different sources resolving logical heterogeneity between the ontologies used by the systems.

1.1 Objective

Starting from works on building mediators for e-Learning resources ([6,7], we show how e-Learning systems that use different metadata schemes or ontologies can interoperate: by specifying *correspondences* ([8,9]) describing semantical relations between these ontologies. They can be used to transform the metadata from one ontology to another.

In an MBIS such correspondences can be used to integrate sources with a different ontology as the mediator, particularly sources of related domains that are based on other metadata standards. The explicit specification of correspondences allows us to integrate and change sources during the runtime of the system – a prerequisite for evolving federations of autonomous systems.

In particular, we show how we use correspondences in MIWeb@e-Learning[4] ([7]), a mediator-based information system that provides information for students and teachers about electronically available learning resources and related publications in the web. The mediator schema is based on the (domain-specific) Learning Object Metadata (LOM) standard and its RDF binding. To bridge the heterogeneity between different schemes, the system contains a mapper component that is able to transform RDF data on the basis of correspondences.

The remaining paper is structured as follows: Chapter 2 gives an overview on the MIWeb@e-Learning system. Chapter 3 discusses the integration of heterogeneous ontologies using a mapper component based on correspondences. Then, Chapter 4 shows other scenarios of interoperating e-Learning systems where correspondences and such a mapper component can be used. Finally, Chapter 5 summarizes our experiences on applying mediator concepts on the context of e-Learning resources in the web and identifies challenges for future work.

1.2 Related Work

Mediator-wrapper architectures in the context of the Web are used in several projects ([10,11]), mainly addressing the problem of query processing. In con-

[3] `http://dublincore.org/`

[4] MIWeb = *M*ediator-based *I*ntegration of *Web* Sources.

trast, we focus on solving heterogeneity using correspondences. Research on explicitly specifying correspondences starts in the beginning of the 90th based on works classifying heterogeneity conflits ([12,5]). The focus was on the definition of languages for correspondences and their use in different contexts like federated information systems, schema integration, and schema mapping ([13,14,15]).

According to the application context we can distinguish mapping approaches that use transformations or queries to define unidirectional mappings between schemes and relational approaches defining a correspondence relationship useable for a transformation in both directions (if possible). We follow a relational approach that can be used for data transformations in both directions. Thereby, we follow a 'view-as-view' approach that is less restricted than 'global-as-view' or 'local-as-view' approaches ([2]).

In the last decade the problem of (semi-)automatic correspondence discovery by ontology matching has moved into the center of research ([16,17,18]). We assume a manual specification for our purposes but hope that further standardisations will reduce the effort for correspondence specification so that they are mainly important to relate resources of different domains (where automatic approaches are difficult to apply).

2 MIWeb @ e-Learning

The MIWeb system[5] provides information on e-Learning resources. Consequently, it integrates web sites of this specific domain: NE (NewEconomy)[6] and DBS (Database Systems) are sources containing metadata of learning objects in the information technology field. In addition, MIWeb automatically connects this information to related web documents using more general metadata sources: the scientific citation index Citeseer[7], the book store of Amazon[8] and the search engine Google.[9] This way, almost any web document can be searched and interpreted from the e-Learning point of view.

The integration of these sources follows a mediator architecture (see Figure 1). It consists of three main components:

- The *mediator* offers the access point to the MIWeb system. It provides a read-access to the integrated sources based on the Learning Object Metadata standard (LOM) [1]. Users can query the system using an SQL like query language. The mediator is responsible for answering these queries using the wrapper components. After collecting all pieces of information, it integrates them into a single result (including the elimination of redundancies).

  The mediator uses descriptions of the wrapper interfaces – the *query capabilities* that are used to dynamically integrate new data sources.

[5] In the following we use MIWeb as the abbreviation of MIWeb@e-Learning.

[6] The New Economy project was founded of the bmb+f within the program 'Neue Medien in der Bildung', `http://neweconomy.e-learning.fu-berlin.de/`

[7] `http://citeseer.ist.psu.edu/`

[8] `http://www.amazon.com/`

[9] `http://www.google.de/`

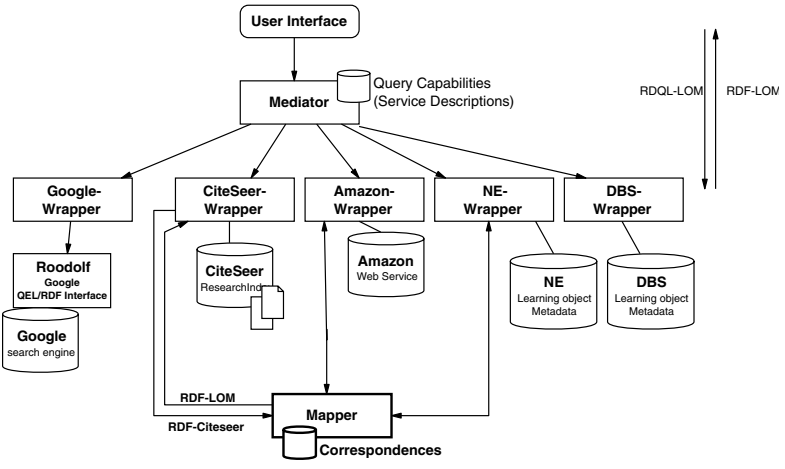


Fig. 1. Architecture of MIWeb@e-Learning

- Each source is encapsulated by a specific *wrapper* component in order to make it compatible to the demands of the mediator: The wrapper must be queryable by the query language defined by the mediator and must provide the source's data according to the LOM metadata standard. For the latter task it uses the mapper component.
- The *mapper* is responsible for *resolving semantical and structural heterogeneity* between mediator and wrapper. In our system it is used to transform data of a source-specific ontology into LOM-compatible data.

  The transformation is based on *correspondences* or *mappings*. Similar to the query capabilities of the mediator, such mappings are specified explicitly to allow changes and extensions. We will describe this component in detail in the next section.

The MIWeb system is based on the Resource Description Framework (RDF) [19] as its common data model and uses the RDF Data Query Language (RDQL)[10] as its query language. RDQL is based on SquishQL and uses SQL-like query constructs.

Mediator Ontology

The mediator of our system uses the Learning Object Metadata Standard (LOM, [1]) as its basic ontology, but extends it to integrate some information about publications. There exists no metadata attribute to specify the number of citations of a publication – an important metadata to estimate its quality. We use the 'Annotation' facility of the LOM standard to add this information to a metadata description.

[10] RDQL is being developed by the Hewlett Packard Semantic Web Group, see `http://www.hpl.hp.com/semweb/rdql.htm`

So, the mediator ontology of the MIWeb system comprehends all documents as learning resources that can be described with a learning object metadata set. Thus, all documents can be treated the same way and it is easy to connect them.

For representation of our mediator schema in RDF we follow the RDF representation of the LOM standard that is described in [20]. It uses top-level metadata standards like the Dublin Core if possible to facilitate the interoperation with them.

3 Model Correspondences for Mapping RDF Data

Model correspondences are useful to integrate heterogeneous ontologies or schemas. Thereby 'heterogeneous' means differences in data models, semantical heterogeneity like naming and domain conflicts, and schematical or structural heterogeneity caused by different representations ([5]).

Whereas data model heterogeneity only depends on the data model concepts, it is a real challenge to identify semantic relationships between ontologies. These *model correspondences* are usually defined manually by a domain expert.

Briefly, a model correspondence is a set of correspondences that specifies data elements of two ontologies that represent the same information. It can be specified either as a relation or as a (uni-directional) mapping like in Clio ([21]) or in XSLT. We propose the specification of correspondences as relations for two reasons: Firstly, the specification of the domain expert can be used for data transformation in both directions. Secondly, we can use the specification for the transformation of data as well as for the transformation of queries. So, we distinguish the terms 'correspondence' and 'mapping':

Correspondence: The corresponding elements provide (at least partially) the same information.

Mapping: A mapping is a computable procedure that transforms values according to a correspondence.

3.1 Structure of RDF Model Correspondences

Figure 2 shows the metamodel for model correspondences. A *model correspondence* relates two ontologies that can be identified by their names. The model correspondence contains a set of *correspondences*. We distinguish simple and complex correspondences. A *simple correspondence* relates two single ontology elements that are identified by their property path starting at the resource they belong to. Thereby we distinguish three types:

- Both the property paths and their values are equal (*SameAs*);
- the property paths are different (*ElementCorrespondence*) or
- property paths and values are different (*ValueCorrespondence* with the additional specification of the corresponding values).

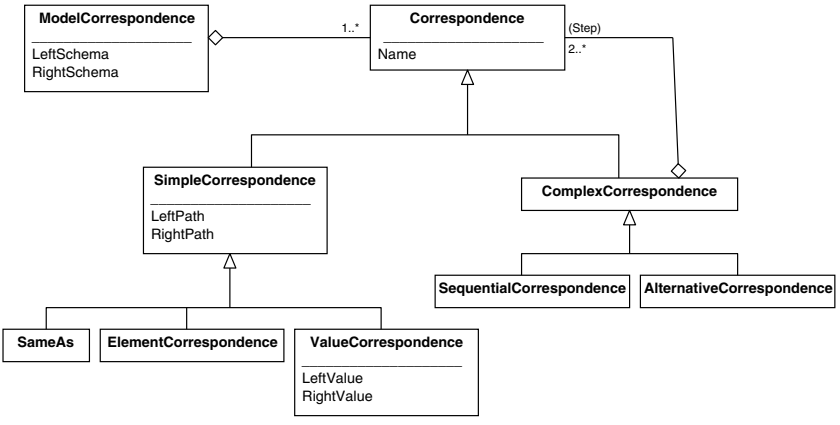


Fig. 2. Structure of Model Correspondences

A *complex correspondence* aggregates a set of correspondences that are needed for a transformation. The included correspondences can be

- *alternatives* that can be used to get the target element (and its values). Then, the related schema represents the same thing in different ways.
- *a sequence* of mapping steps that have to be done step by step to get the target element from the source schema.

Obviously, an alternative correspondence can only be used in one direction. But you can define one alternative as a default mapping that is used for the transformation to the schema with more alternatives.

The metamodel is specified by an XML schema. It contains constraints like cardinalities and required attributes we omit here - it can be found in [7].

3.2 Examples

In the following we give some examples of correspondences between the New Economy source and the MIWeb mediator. Thereby, we show the data structures from the RDF data of the related ontologies and characterise the type of correspondence. In particular, we can see

- how same structures are defined with same as correspondences,
- how different attribute domains can be related with value correspondences (Aggregation Level),
- how aggregations of elements into bags or other collections can be done using an element correspondences (see Learning Resource Type), and
- how differences in one source can be handled by alternative correspondences (see the 'is part of' relationship).

1.2 General.Title	same as correspondence
New Economy / Mediator	
`<dc:title>Analysis...</dc:title>`	

1.8 General.Aggregation Level	value correspondence
New Economy	Mediator

`<ne:granularitytype rdf:resource=".../NE/rdfs#component"/>`
 	
`<ne:granularitytype rdf:resource=".../NE/rdfs# physicalelement"/>`
	`<lom-gen:aggregationLevel rdf:resource=".../lom-general# AggregationLevel2"/>`
	
	`<lom-gen:aggregationLevel rdf:resource=".../lom-general# AggregationLevel1"/>`

5.2 Educational.Learning Resource Type	element correspondence
New Economy	Mediator

`<rdf:type>`
`<rdf:Bag>`
`<rdf:li>.../lom-educational #Introduction</rdf:li>`
`<rdf:li>.../lom-educational #NarrativeText</rdf:li>`
`</rdf:Bag>`
`</rdf:type>`
	`<rdf:type rdf:resource= ".../lom-educational#Introduction"/>`
	`<rdf:type rdf:resource= ".../lom-educational#NarrativeText"/>`

7.2 Relation.Resource.Identifier for 7.1 Relation.Kind = is part of	alternative correspondence with two element correspondences
New Economy	Mediator

`<dcterms:isPartOf rdf:resource="http://.../index.html"/>`
 	
or
 	
`<dcterms:isPartOf>`
`<rdf:Bag>`
`<rdf:li>http://.../index.html</rdf:li>`
`<rdf:li>http://.../a31.html</rdf:li>`
`</rdf:Bag>`
`</dcterms:isPartOf>`
	`<dcterms:isPartOf>`
	`<rdf:Bag>`
	`<rdf:li rdf:resource="http://.../index.html"/>`
	`<rdf:li rdf:resource="http://.../a31.html"/>`
	`</rdf:Bag>`
	`</dcterms:isPartOf>`

3.3 The Mapper Component

The mapper supports three tasks: the transformation of RDF data, the management of correspondences, and the administration of the mapper component itself. As the transformation algorithm is specific for each kind of correspondence, we use specific Rule classes realizing the transformation operation for one specific Correspondence type. This design allows the integration of new rules if the metamodel is extended.

To transform RDF data, the mapper firstly looks for a model correspondence between the given models using the correspondence registry. It then identifies the direction the correspondence has to be interpreted. For each contained correspondence, the transformer tries to find a rule that can do the transformation. If no rule can be found, i.e. there exists no transformation algorithm for this correspondence type yet, the correspondence is skipped. Similar, errors during the transformation will not break off the transformation process. So, we follow the idea of transforming as much data as possible. The result of the transformation as a whole is the union of all the RDF models built from the rules.

4 Other Interoperation Scenarios

Correspondences used by a mapper component are a flexible approach for several interoperation scenarios of e-Learning systems. We briefly describe four of them (see Figure 3):

Multi-ontology-based Mediation. The MIWeb system introduced in the last sections follows a classical mediator architecture with only one mediator ontology. By integrating a mapper component into the mediator with predefined correspondences between several existing standards in the e-Learning domain, the integration of a source will be much easier: It only has to reference the standard used by this particular source. Similar, the user can query the resources using the standard he knows.

P2P-based Interoperation. The mediator-wrapper architecture associates specific tasks to the system components. If we allow an interoperation of e-Learning systems using components with the same functionality, we get a P2P-based interoperation. Each peer must have the knowledge how to interoperate with peers it knows. So, each peer will use a mapper component with the correspondences to the ontologies of connected peers.

Message-oriented Interoperation. Using a centralized approach as in EAI systems, a message broker is responsible for transforming messages (like e-Learning metadata) between interoperating systems. Thus, the broker can use a mapper component configured with appropriate correspondences.

Direct Interoperation. If there exists a pre-defined information flow on a few e-Learning systems, this can also be supported by transformations based on a correspondence. Even a migration scenario is an example for such a situation. Usually the mapper component does not provide a registry interface but only the transformation service.

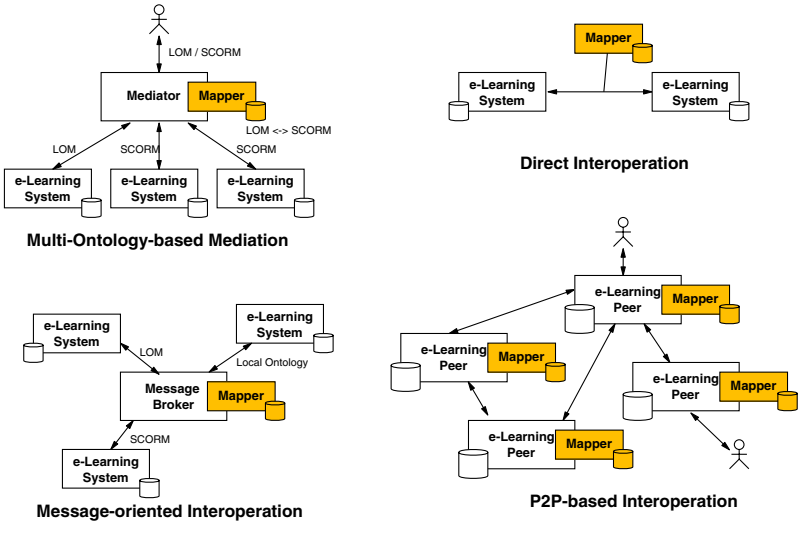


Fig. 3. Interoperation Scenarios using Correspondences

5 Conclusions

The MIWeb@e-Learning system shows how concepts of federated information systems can be used to build a domain-specific mediator system about e-Learning resources and related publications. It uses web technology and RDF for integration. To bridge the heterogeneity between different ontologies we propose to use metadata standards and *correspondences* describing semantic relationships between them. The explicit specification and management in a mapper component allows one to change wrappers and its ontologies dynamically. This way, the autonomy of e-Learning systems as well as future ontology changes are taken into account. Furthermore, the implementation can be reused to build mediators for other domains. In contrast to the classical mediator-wrapper-architecture we define an explicit mapper component for the transformation. It can be used in many different scenarios that require the interoperation of e-Learning systems.

In detail, we made the following experiences with metadata standards and the technologies usually used within the web context:

- The RDF format and the related RDF schema are well suited for semistructured web data and its describing metadata. In particular, the reification mechanism allows a mediator system to document its integration.
  There exist many query languages for RDF ([22]). We used RDQL because it is very readable since it is quite similar to the notation of SQL. But our experiments have shown that it also has disadvantages in our distributed scenario: We often want *any* information that we could get, even if some of the required attributes are missing. But such queries require a language that allows optional pathes like SPARQL ([23]).
- The LOM standard and its RDF binding are well suited for a mediator ontology. The standard defines both attributes and their domains - a good basis for the integration. The RDF binding shows explicitly the relationships to top level standards like the Dublin Core and thus supports interoperability to more general descriptions.

Besides the improvement of correspondences and the mapper component implementation our future work will addresss two aspects for the interoperation, particularly the search of e-Learning resources:

- Query relaxation: Domain-specific ontologies allow an expert search with many attributes. In our experience it can be difficult to specify a query to retrieve appropriate results. Thus, more investigations are necessary to the specification process of queries, for example with query relaxation ([24]).
- Pre-defined correspondences between ontologies of the e-Learning domain: Standardized ontologies are a first step to interoperable e-Learning systems and a good basis for defining correspondences independent from specific systems. This way, the effort needed for the specification of correspondences will be reduced.

References

1. IEEE Learning Technology Standards Committee: Standard for Information Technology – Education and Training Systems – Learning Objects and Metadata. Technical report, IEEE (2002)
2. Ullman, J.: Information Integration using Logical Views. In: Proc. 6th Int. Conf. on Database Theory, Delphi, Greece (1997)
3. Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hübner, S.: Ontology-Based Integration of Information – A Survey of Existing Approaches. In: IJCAI-01 Workshop 'Ontologies and Information Sharing'. (2001) 108–117
4. Wiederhold, G.: Mediators in the Architecture of Future Information Systems. In Huhns, M.N., Singh, M.P., eds.: Readings in Agents. Morgan Kaufmann, San Francisco, CA, USA (1997) 185 – 196
5. Busse, S., Kutsche, R.D., Leser, U., Weber, H.: Federated Information Systems: Concepts, Terminology and Architectures. Forschungsberichte des Fachbereichs Informatik Nr. 99-9, Technische Universität Berlin (1999)
6. Sigel, W., Busse, S., Kutsche, R.D., Klettke, M.: Designing a Metadata-based Infrastructure for E-Learning. In: Proc. of the 5th Workshop Engineering Federated Information Systems (EFIS). (2003) 89 – 99
7. Busse, S., Kabisch, T., Petzschmann, R.: Mediator-based Integration of Web Sources (MIWeb) - Case Study e-Learning. Forschungsberichte der Fakultät IV - Elektrotechnik und Informatik Nr. 2005-02, Technische Universität Berlin (2005)
8. Leser, U.: Query Planning in Mediator Based Information Systems. PhD thesis, TU Berlin, Fachbereich Informatik (2000)
9. Busse, S.: Modellkorrespondenzen für die kontinuierliche Entwicklung mediatorbasierter Informationssysteme. PhD thesis, TU Berlin, Fakultät IV Elektrotechnik und Informatik (2002) Logos Verlag Berlin.
10. J.Petrini, T.Risch: Processing queries over RDF views of wrapped relational databases. In: 1st Int. Workshop on Wrapper Techniques for Legacy Systems, WRAP 2004. (2004) 16–29
11. Sattler, K., Geist, I., Schallehn, E.: Concept-based Querying in Mediator Systems. VLDB Journal **14** (2005) 97–111
12. Miller, R.: Using schematically heterogeneous structures. ACM SIGMOD Record **27** (1998) 189–200
13. Papakonstantinou, Y., Abiteboul, S., Garcia-Molina, H.: Object fusion in mediator systems. In: VLDB '96: Proceedings of the 22th Int. Conf. on Very Large Data Bases, Morgan Kaufmann (1996) 413–424
14. Levy, A.Y., Rajaraman, A., Ordille, J.J.: Querying heterogeneous information sources using source descriptions. In: VLDB '96: Proceedings of the 22th Int. Conf. on Very Large Data Bases, Morgan Kaufmann (1996) 251–262
15. ISO: Industrial automation systems and integration – Product data representation and exchange – part 11: Description methods: The EXPRESS language reference manual. ISO Standard 10303 part(11), 2nd ed., ISO TC184/SC4 (2004)
16. Castano, S., Ferrara, A., Montanelli, S., Racca, G.: Matching techniques for resource discovery in distributed systems using heterogeneous ontology descriptions. In: Proc. of the Int. Conf. on Coding and Computing (ITCC 2004), IEEE (2004)
17. Doan, A., Madhavan, J., Dhamankar, R., Domingos, P., Halevy, A.: Learning to match ontologies on the semantic web. The VLDB Journal **12** (2003) 303–319

18. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. The VLDB Journal **10** (2001) 334–350
19. W3C World Wide Web Consortium: RDF Primer. W3C Recommendation, REC-rdfprimer-20040210 (2004)
20. Nilsson, M., Palmer, M., Brase, J.: The LOM RDF binding – principles and implementation. In: 3rd Annual Ariadne Conference, Belgium (2003) for a draft of the RDF binding see `http://kmr.nada.kth.se/el/ims/metadata.html`.
21. Popa, L., Velegrakis, Y., Miller, R., Hernandez, M., Fagin, R.: Translating web data. In: VLDB '02: Proc. of the 28th Int. Conf. on Very Large Data Bases, Morgan Kaufmann (2002) 598–609
22. Magkanaraki, A., Karvounareakis, G., Anh, T., Christophides, V., Plexousakis, D.: Ontology Storage and Querying. Technical Report 308, Foundation for Research and Technology Hellas, Information Systems Laboratory (2002)
23. W3C World Wide Web Consortium: RDF SPARQL Query Language for RDF. W3C Working Draft, WD-rdf-sparql-query-20050721 (2005)
24. Lee, D.: Query Relaxation for XML Model. PhD thesis, University of California, Los Angeles (2002)

Towards Ontology-Guided Design of Learning Information Systems*

Aldo de Moor

STARLab, Vrije Universiteit Brussel,
Belgium
ademoor@vub.ac.be

Abstract. Courseware increasingly consists of generic information and communication tools. These offer a plethora of functionalities, but their usefulness to a particular learning community is not easy to assess. The aim should be to develop comprehensive learning information systems tailored to the specific needs of a community. Design patterns are important instruments for capturing best practice design knowledge. Ontologies, in turn, can help to precisely capture and reason about these patterns. In this paper, we make the case for ontology-guided learning IS design, and sketch the ontological core of a potential design method. Such methods should enable communities to specify and access relevant best design practice patterns. Design knowledge can then be reused across communities, improving the quality of communication support provided, while preventing wheels from being reinvented.

1 Introduction

Courseware has become an essential means of course-based, Web-supported e-learning. A plethora of tools and environments exists. On the one hand, there are more or less comprehensive courseware platforms: commercial platforms like WebCT and Blackboard, and open platforms which can be completely or partially open source. However, e-learning is increasingly also being supported by collections of generic collaborative tools, such as blogs and wikis, e.g. [1].

Courseware platforms and tools offer many functionalities which can be used to support communities of users in their individual and especially collaborative needs. Much research and development efforts concentrate on producing increasingly advanced components for knowledge sharing and learning [2, 3]. However, many design and use problems arise as such functionality is becoming more widely available, including un(der)used functionalities, gaps between required and available functionalities, and conflicting requirements of different stakeholders. Such mismatches between collaborative requirements and available functionalities can be characterized as socio-technical gaps [4].

Practical evaluation methods are necessary for effectively and efficiently designing comprehensive and useful learning information systems. These we define as sets of

* This research was supported by the EU Leonardo da Vinci CODRIVE project B/04/B/F/PP-144.339. The author wishes to thank Jurrit de Vries and Jaap Wagenvoort for their help with the experiment described in the example.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1190 – 1199, 2005.
© Springer-Verlag Berlin Heidelberg 2005

interrelated courseware and tool functionalities that satisfactorily serve the information and communication needs of a particular learning community. Many evaluation methods score and weigh the quality of functionalities. However, mere numerical evaluations have no clear semantics, and thus are hard to compare and reuse across learning communities. In order to provide learning communities also with semantic building blocks for designing their information systems, we examine in this paper the role that ontologies can play. To this purpose, we make the case for ontology-guided learning information system design patterns. Design patterns capture the essential lessons learnt by a community about which technologies work for it for what purposes. By formalizing these patterns using ontologies, the effectiveness and efficiency of its design processes, and thus of the resulting information systems, can be improved. In Sect. 2, we present a Social Context Model for communication processes, which provides the core patterns for the ontology. Sect. 3 examines the idea of ontology-guided learning IS design. We sketch the contours of a possible design method in Sect.4. We end the paper with conclusions.

2 A Social Context Model for Communication Processes

E-learning communities are communities of practice, which involve a domain of knowledge defining a set of issues, a community of people who care about this domain, and the shared practice that they develop to be effective in their domain [5]. Such communities require sophisticated system designs to support their communicative needs [6]. The problem with current analysis approaches is that they often classify communication technologies only in terms of the basic communication moves they afford, such as the creation of issues or arguments pro/contra. Needed, however, is a way to analyze collaborative technologies and the communication processes they enable in their full *context of use*. While a mailing list may work perfectly fine as a coordination tool in a small research community, it generally works badly as a tool to coordinate the multiple assignments given to many groups in a particular course. Another example is a divergent discussion process, which may be very useful in a friendly social community, but quite insufficient when used to support official document authoring processes with large interests at stake.

In [7, 8], we developed a Social Context Model (SCM) for the analysis of the functionalities provided by communication tools. The model helps to position communication technologies and processes in terms of their usefulness for particular communication purposes. The model examines communication processes from a process *context* dimension consisting of communication process layers, and from a process *structure* dimension describing the configuration of the elements of the communication processes.

The SCM consists of four layers of communication processes, each higher level process providing a context that embeds the lower-level processes. From high to low-level processes these are: collaboration, authoring, support, and discussion processes:

- **Collaboration processes** *give purpose* to the collaboration activities, discussions, and documents.
- **Authoring processes** *produce* the structured outputs of the collaboration processes, such as documents.

- **Support processes** focus on the *organization* of the discussions between the participants, ensuring that they contribute to the creation and interpretation of the output.
- **Discussion processes** are the actual conversational exchanges in which the argumentation between participants takes place.

Each communication process, whether it is a discussion process or one of its embedding context processes, has a structure comprised of certain process entities. First, there are the *process elements* (the elements a process itself is made of, such as goals, roles, and objects). Second, there are the processes constructed out of these elements. These we subdivide into *actions* (which constitute the actual communication workflows) and *change processes* (meta-processes in which the communication process can be adapted).

In both actions and change processes, there are *norms* that describe the acceptable behavior in the community, by defining the authorizations of the participants in the process roles that they play. All communities have such norms, either explicitly laid down in charters and by-laws, or only implicitly defined, but no less strong in impact [9, 10]. These context-grounded norms are the core elements needed to see which collaborative patterns may, must, or may not be present in a particular socio-technical system, and are thus essential in systems design.

We have applied the SCM in various ways. One use is to chart collaborative requirements in some community of practice. Another application is comparing the usage contexts of collaborative functionalities afforded by various tools. Once these data are known, matches between requirements and functionalities can be performed to identify socio-technical gaps, rank tools in terms of usefulness, etc.

	Process Elements *(Goals, Roles, Objects)*	**Actions** *(Workflows, Norms)*
Collaboration Proc. *(Why?)*		
Authoring Proc. *(What?)*	• Author • Editor	• Author – May – Add_Position_to_Section • Author – May – Take_Position_in_Section • Editor – Must – Edit_Section_Introduction • Editor – Must – Edit_Section_Conclusion
Support Proc. *(How organized?)*		
Discussion Proc. *(How done?)*	• Discussant • Position • Argument	• Discussant – May – Create_Position • Discussant – May – Add_Argument

Fig. 1. An SCM Affordance Analysis of GRASS

An example of an examination of the affordances provided by a collaborative tool, is an analysis of the GRASS (Group Report Authoring Support System) tool, which allows adversarial stakeholder communities to assess the amount of consensus they

have reached on conflict issues [11]. Fig.1 shows a simplified analysis of this example (amongst other things not taking into account the change processes). It shows that GRASS is especially strong in typical issue-network functionalities (creating positions and arguments pro/contra positions or other arguments). In addition, it allows these issue-networks to be part of group reports, by permitting authors to add positions (plus their associated argumentation trees) to particular sections, and to take positions as desired. These norms are privileges (permitted actions). Examples of responsibilities (required actions) are that editors *must* create section introductions and conclusions. The analysis also shows that two types of communication processes are *not* supported by GRASS at the moment: support processes (how to organize discussions, for example facilitation or summarization of discussions) and collaboration processes (the purpose of the reports produced, e.g. societal conflict resolution or class assignments).

Although the SCM has proven its value in practice, a limitation is still that so far it only has been used for informal, qualitative analysis. Its usefulness would increase, however, if part of its representation and associated analysis processes are formalized. One way to do so is by using the SCM as the basis for ontology-guided design of socio-technical systems.

3 Ontology-Guided Learning IS Design

Ontologies are useful instruments for improving the reusability of collaborative knowledge. An ontology is a shared conceptualization of an application area that defines the salient concepts and relationships among concepts, and as such can assist in modeling and implementing solutions to application tasks [12]. Three terms are especially important in this definition: *shared*, *conceptualization,* and *application*. In order to be *shared*, there needs to be a process in which conceptualizations get agreed upon. The *conceptualization* itself requires some formalism, in the sense of knowledge representation plus reasoning procedures. Finally, this shared conceptualization should contribute to solving an *application* problem.

Reaching shared meanings is not trivial, since community members may have different interests and different visions of the domain under consideration [13]. The difficult social process of sharing and reaching consensus on joint meanings is an important area of ontological research [14, 15]. However, here we focus on the conceptualization itself, with the application area being design of learning information systems.

In the design of complex (socio-technical) systems, design patterns are very important. Humans are very good at using patterns to order the world and make sense of things in complex situations [16]. A pattern is a careful description of a perennial solution to a recurring problem within a […] context (Alexander, in [17]). Thus, patterns are context-dependent and relatively stable. They need to be concrete enough to be useful, while also being sufficiently abstract to be reusable. Such patterns can be captured and used in conceptual models like the SCM and ontology-based knowledge systems.

In our work, we have adopted conceptual graphs to model and use design patterns [18]. Conceptual graphs are a flexible and extensible method for knowledge

representation [19, 20]. Their strength is not in the range of semantic constraints they can represent, other formalisms such as ORM provide much richer semantic constraint primitives [21]. What makes conceptual graphs particularly useful for pattern analysis are the generalization hierarchies, not only of concept and relation types, but also of complete graphs. The formalism contains a set of powerful operations such as projections, generalizations, and joins that make use of the properties of the graph hierarchies and their components. Furthermore, they not only support the description of complex domain knowledge, but also the validation of that knowledge against meta-knowledge about the domain (such as quality criteria that domain graphs have to match with). Thus, conceptual graphs are well suited to represent and reason about pattern and context knowledge.

Next, we outline some requirements and elements of a possible approach, using examples from a learning IS design case.

4 Towards a Design Method

There is no such thing as THE design choice in a community of practice. Each community has its own requirements, over time developing its own unique perspectives, common knowledge, practices, supporting technologies, approaches, and established ways of interacting [5]. Thus, to properly support a community in its systems development, design patterns need to evolve over time as well, generally becoming more specialized and tailored. Yet if communities are so unique, how then can their design best-practices be shared?

We illustrate the design problem and the use of conceptual graphs as the core of a potential design method by examining material from a real learning IS design case. The main point we want to make in this paper is that ontology-based design is needed to create better learning information systems. Our aim is not not present a full method for ontology-guided learning information system design. Both the learning problem and conceptual graphs-based ontological analysis presented are much simplified examples of what is needed in practice. However, in an educational technology field predominantly focusing on standards and technology platforms, the case should illustrate that a systematic, formal semantics-based approach to learning information *systems*-design is an important part of the puzzle as well.

4.1 An Example: Refining Notification Patterns

Providing computer support for the synchronization of asynchronous collaborative learning activities is vital for learning communities [22]. For example, an important synchronization functionality is notification of events through e-mail. As soon as a document is added to a particular folder, the members of a workspace often receive an e-mail, including a clickable link so that they can view the change immediately.

Although conceptually simple, developing appropriate notification support by a learning information system requires answering many questions: who is to be notified? All users, or just a relevant subset? How to determine to which users a particular change is relevant? How often to notify users: with every event, daily,

weekly? Which technology to use: a 'push' technology like e-mail, or a 'pull' technology like some web tool?

This issue came to the fore during an experiment with building and using a learning information system consisting of a set of generic information tools. In the Fall 2004 semester, 15 Information Management students taking a course given by the author on 'Quality of Information Systems' were given the assignment to jointly write a report on a societal theme. To this purpose, they were to use a set of blogs to collect and interpret source materials for the report, and use the GRASS tool mentioned earlier to write the report. Since discussions were scattered over many fora, we initially adopted the design pattern 'send a notification whenever some discussion process takes place' (Fig.2a). To our great surprise, of 13 students having participated in the evaluation held after the experiment, 10 'completely disagreed' and 3 'disagreed' with the proposition that the notification of changes in the report was valuable. This suggested that the general notification design pattern of Fig.2a should be discarded in a student population having the goal of writing a group report. Instead, the correct pattern seemed to be that no notification should be sent at all in case of an addition to the discussion. Still, this design pattern also seemed too extreme: to a follow-up question 7 students answered they did not want to receive any notification at all, but 2 still preferred a daily, and 4 a weekly digest. This, plus the fact that there had been some technical problems resulting in a flood of notification mails at one point in time, suggest that it was not so much the notification functionality itself that was problematic, but its high frequency and indiscriminate nature. An alternative design, therefore, could be to add an entry about any discussion event to a Web-accessible change log, while still sending each student an e-mail about rare, 'quality'-events, such as the posting of a discussion summary by an editor (Fig.2b).

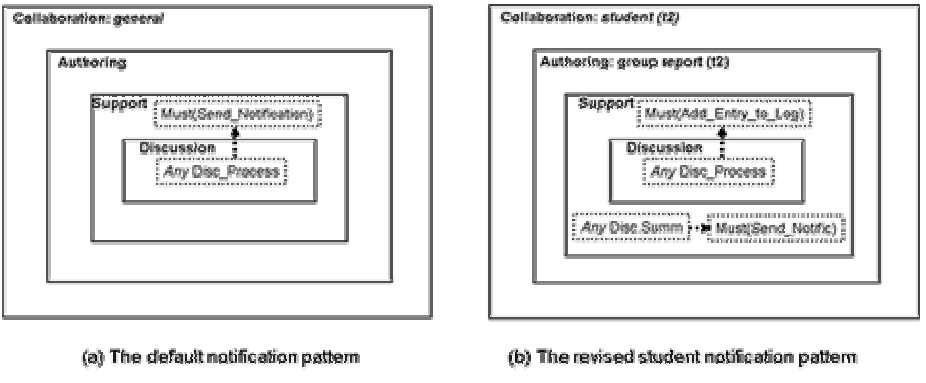


Fig. 2. Notification Pattern Refinement

4.2 Pattern Analysis with Conceptual Graphs

This example shows how design patterns can evolve in practice. It is important to note that both requirements (e.g. practices of students working on group reports) and the (mappings to) supporting functionalities (e.g. notification process functionality)

become more specialized over time. Ontologies in general, and conceptual graphs in particular, allow for these patterns to be efficiently analyzed and reused when embedded in appropriate design methodologies. To show proof of principle, we now explain how the above example could be formalized using conceptual graphs.

Conceptual graphs are constructed out of concepts and relations. A concept has two fields: a type and a referent field. The type-field contains a type label that is part of a concept type hierarchy. The referent-field designates a particular entity with the mentioned type. This field is optional: if not specified, the concept is considered to be generic, and is by default existentially quantified. A conceptual relation links two or more concepts. Each conceptual relation has a relation type. It also has one or more arcs, represented by arrows, each of which must be linked to some concept. A conceptual graph is a combination of concept nodes and conceptual relation nodes. It can also consist of a single concept node.

Generalization/specialization semantics are a core feature of conceptual graphs. Besides concept type hierarchies, also a relation type hierarchy and a generalization hierarchy of graphs are distinguished. This enables powerful abstraction and comparison operations to be carried out, which are core building blocks of pattern analysis.

Returning to the notification example: the design patterns identified there can be easily translated into conceptual graphs. For example, Fig.3 contains the (partial) conceptual graph representation and associated concept type hierarchy of the informal notification design pattern depicted in Fig.2c. The key concept of the type hierarchy is the STS_Pattern (socio-technical system-pattern). The hierarchy contains the four main types of SCM communication processes: Collaboration, Authoring, Support, and Discussion processes. A Deontic_Effect can be attached to concepts to indicate their normative status (i.e. required, permitted, forbidden). Only one type of Actor is currently distinguished, namely Student, while two types of functionalities are described: Send_Notification and Add_Entry_to_Change_Log. Three objects are Group_Report, Discussion_Summary, and Discussion_Process. The latter is labelled as an object, since in the system it is the representation of the discussion process (for example an addition to a discussion thread) that, for instance, triggers the addition of an entry to the change log. Of course, the discussion process as a *process* also needs to be of interest and included in the type hierarchy, but is not shown here to reduce the complexity of the example.

The notification pattern graph that is built on top of the semantics of this type hierarchy shows, for example, that a Support-process exists in which a Discussion_Summary-object being added triggers the Send_Notification-functionality, and that the latter has an associated Deontic_Effect (i.e. normative status) of being required. Note that the (agent, part, result, and characteristic) relations are often used in conceptual graph theory, but other, community-defined relations can be used just as well.

Having represented design patterns is only the first step. The second step is to use these representations in the design process. One way is to develop *query graphs*: using the standard conceptual graph-*projection* operation, all graphs from a set of socio-technical systems design patterns that are specializations of the query graph will be returned. This can help a community in reusing existing ideas for its own design patterns.

In [18], we present a more elaborate method based on socio-technical systems patterns that allows for collaboratory improvement. A similar method could be developed for learning information systems design.

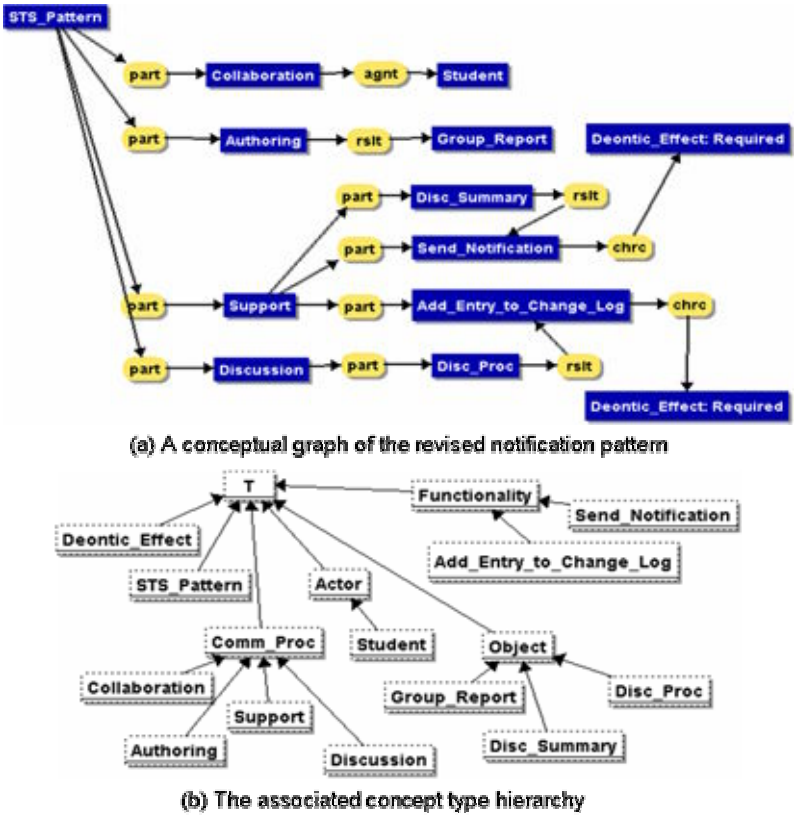


Fig. 3. A Conceptual Graph of the Notification Design Pattern

5 Conclusions

Learning information systems, meaningfully constructed out of constellations of courseware platforms and generic communication tool components, will play an increasingly important role in supporting learning communities. Design patterns are useful instruments in facilitating the community-driven systems development process. Although useful, informal design patterns such as based on the Social Context Model, are not sufficient when reusability of patterns is a goal. Ontologies, in the form of shared, formalized conceptualizations of a domain and application area, can help promote reusability and usefulness of such patterns.

The main point of this article was to show the need for ontology-guided design of learning information systems. We only gave a brief sketch of a possible method for

supporting this process using conceptual graphs, and did not give more details about the particular ontological approach to use, since that was not the point. In future work, at the very least, an extension of the SCM concept (and relation) type hierarchy is needed, like more refined goals, roles, objects, workflows, and norms, although the particular extensions will depend on the (category of) learning community. By building on a common ontological core, each community can define its own semantics, while still allowing for comparisons with other patterns to be made. Ultimately, such conceptualizations should become full (learning IS) pattern languages that allow communities to express rich collaboration requirements, i.e. along the lines of the Pattern Language for Living Communication described in [17]. Such a language, in turn, is the starting point for the design of a proper community design *methodology,* which was outside the scope of this paper but equally important.

To capture and reason about the formalized patterns, conceptual graphs are not the only formalism to be used. Related ontology formalisms and methods are extensively being researched in the Semantic Web and business process modeling research communities, e.g. [23]. Multiple knowledge representation formalisms can and should complement and enrich each other [24]. One of our research objectives therefore is to extend the powerful pattern comparison feature of conceptual graphs with the rich semantic constraints provided by Object-Role Modeling [21]. This research will take place in the context of the DOGMA approach to ontology engineering, developed at STARLab [25]. This approach aims to satisfy real world needs by developing a useful and scalable community-grounded ontology engineering approach, allowing for much more subtle representation and analysis of design patterns. Such a combination of a pragmatic ontological engineering methodology informed by high-quality, learning community defined design patterns, could open the road to much more customized, high-quality learning information systems.

References

1. Efimova, L. and S. Fiedler. Learning Webs: Learning in Weblog Networks. in *Proc. of the IADIS International Conference on Web Based Communities 2004, Lisbon, Portugal*. 2004: IADIS Press.
2. Roschelle, J., et al., Developing Educational Software Components. *IEEE Computer*, 1999. 32(9): p. 2-10.
3. Bieber, M.e.a. Towards Knowledge-Sharing and Learning in Virtual Professional Communities. in *Proc. of the 35th Hawaii International Conference on System Sciences, Hawaii, January 5-7, 2002*. 2002.
4. Ackerman, M.S., The Intellectual Challenge of CSCW: the Gap Between Social Requirements and Technical Feasibility. *Human-Computer Interaction*, 2000. 15(2): p. 179-203.
5. Wenger, E., R. McDermott, and W. Snyder, *Cultivating Communities of Practice*. 2002, Cambridge, MA: Harvard Business School Press.
6. Etzioni, A. and O. Etzioni, Face-to-Face and Computer-Mediated Communities: A Comparative Analysis. *The Information Society*, 1999. 15: p. 241-248.
7. de Moor, A. and R. Kleef. Authoring Tools for Effective Societal Discourse. in *Proc. of the 15th International Informatics for Environmental Protection Symposium: Sustainability in the Information Society*. 2001. Zurich, Switzerland.

8. Kleef, R. and A. de Moor, Communication Process Analysis in Virtual Communities on Sustainable Development, in *Environmental Online Communication*, A. Scharl, Editor. 2004, Springer: Berlin.
9. Preece, J., *Online Communities : Designing Usability, Supporting Sociability*. 2000, Chichester ; New York: John Wiley.
10. Wershler-Henry, D. and M. Surman, *Commonspace: Beyond Virtual Community*. FT.Com Books. 2001, Toronto: Pearson.
11. Heng, M. and A. de Moor, From Habermas's Communicative Theory to Practice on the Internet. *Information Systems Journal*, 2003. 13(4): p. 331-352.
12. Musen, M.A., Ontology-Oriented Design and Programming, in *Knowledge Engineering and Agent Technology*, J. Cuena, et al., Editors. 2000, IOS Press: Amsterdam.
13. Gruninger, M. and J. Lee, Ontology: Applications and Design. *Communications of the ACM*, 2002. 45(2): p. 39-41.
14. Euzenat, J., Building Consensual Knowledge Bases: Context and Architecture, in *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, N. Mars, Editor. 1995, IOS Press: Amsterdam. p. 143-155.
15. Edgington, T., et al., Adopting Ontology to Facilitate Knowledge Sharing. *Communications of the ACM*, 2004. 47(11): p. 85-90.
16. Kurtz, C.F. and D.J. Snowden, The New Dynamics of Strategy: Sense-Making in a Complex and Complicated World. *IBM Systems Journal*, 2003. 42(3): p. 462-483.
17. Schuler, D. A Pattern Language for Living Communication. in *Participatory Design Conference (PDC'02), Malmo, Sweden, June*. 2002.
18. de Moor, A., Improving the Testbed Development Process in Collaboratories, in *Proc. of the 12th International Conference on Conceptual Structures (ICCS 2004), Huntsville, AL, USA, July 2004*. 2004. p. 261-274.
19. Sowa, J.F., *Conceptual Structures : Information Processing in Mind and Machine*. 1984, Reading, Mass.: Addison-Wesley.
20. Mineau, G.W., R. Missaoui, and R. Godinx, Conceptual Modeling for Data and Knowledge Management. *Data & Knowledge Engineering*, 2000. 33: p. 137-168.
21. Halpin, T., Object-Role Modeling (ORM/NIAM), in *Handbook on Architectures of Information Systems*, P. Bernus, K. Mertins, and G. Schmidt, Editors. 1998, Springer-Verlag: Berlin.
22. Lundin, J., Synchronizing Asynchronous Collaborative Learners, in *Communities and Technologies*, M. Huysman, E. Wenger, and V. Wulf, Editors. 2003, Kluwer Academic: Dordrecht. p. 427-433.
23. Sheth, A. and R.A. Meersman, eds. *Proc. of the NSF-EU Workshop on Database and Information Systems Research for Semantic Web and Enterprises, April 3-5, Amicalola Falls and State Park, GA, USA*. 2002.
24. Markman, A.B., *Knowledge Representation*. 1999, London: Lawrence Erlbaum.
25. Spyns, P., R.A. Meersman, and M. Jarrar, Data Modelling versus Ontology Engineering. *ACM SIGMOD Record*, 2002. 31(4): p. 12-17.

Learning to Generate an Ontology-Based Nursing Care Plan by Virtual Collaboration*

Woojin Paik[1] and Eunmi Ham[2]

[1] Dept. of Computer Science, Konkuk University,
322 Danwol-Dong, Chungju-Si, Chungcheongbuk-Do, 380-701, Korea
wjpaik@kku.ac.kr
[2] Dept of Nursing Science, Konkuk University,
322 Danwol-dong, Chungju-si, Chungcheongbuk-do, 380-701, Korea
hem2003@kku.ac.kr

Abstract. We describe how a web-based collaboration system is used to generate a document by referring ontology within a specific subject area. We have chosen nursing science as the target subject area as the nurses collaboratively plan and apply necessary treatments to the patients. The planning process involves coming up with a set of decisions and activities to perform according to the knowledge embedded in the nursing subject ontology. The nurses copiously record the patient conditions, which they observed, and also the ensuing reasoning outcome in the form of the diagnoses. Nursing care plan is a representative document, which is generated during the application of nursing processes. The plan includes general patient information, medical history, one or more goals of the nursing care plan, nursing diagnoses, expected outcomes of the care, and possible nursing interventions. We are developing a collaborative nursing care plan generation system, where several nurses can record and update the collected factual information about the patients and then come up with the most appropriate nursing diagnoses, outcomes, and interventions. The nursing care plan generation system is designed to double as a learning aid in order for the nurses by allowing them to observe what others do during the plan generation process. Eventually, the nurses are expected to share the same semantics of each ontology as they repeat the ontology-based decision makings.

1 Motivation: Nursing Care Plan

Nursing care plan is an embodiment of how nurses apply a clinical reasoning process to analyze and evaluate gathered facts about patients. The nursing care plan is the source of knowledge involved in the overall nursing process, which leads up to the nursing diagnosis. The nursing care plan includes the systematic explanation of the facts gathered from the patient assessment stage and also all intervening analysis by the nurses as well as the final diagnosis, interventions to be performed, and expected outcome of the interventions. The nurses develop a plan of care that prescribes interventions to attain the expected outcomes. The care plan is prepared to provide continuity of care from nurse to nurse, to enhance communication, to assist with

* Corresponding author.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1200 – 1204, 2005.
© Springer-Verlag Berlin Heidelberg 2005

determination of agency or unit staffing needs, to document the nursing process, to serve as a teaching tool, and to coordinate provisions of care among disciplines [1].

Typical nursing care plan includes the background information about the patients such as the general biographical information, the medical history, various health related information, physical and mental state assessment results, nursing diagnoses, suggested interventions, and expected outcomes. Often certain information such as the medication records is entered in a tabular form. However, much of the information is conveyed as narratives of the nurses and the direct quotes from the patients. In the nursing process literature, patient data are regarded as either subjective or objective. Subjective data are what the client reported, believed, or felt. Objective data are what can be observed such as vital signs, behaviors, or diagnostic studies [1]. The nurses record patient's condition in the form of facts that the nurses observed, monologue by the patient, patient's answers to the nurse's questions, and the patient's condition conveyed by others such as family members. Then, the nurses often summarize the factual information. The nurses form their evaluation of the patient's condition based on the summarized factual information.

The ultimate goal of the nursing process is to come up with a nursing diagnosis, which can guide the selection of the most suitable interventions and the expected outcomes. According to North American Nursing Diagnosis Association (NANDA), a nursing diagnosis is defined as a critical judgment about individual, family, or community responses to actual or potential health problems or life processes. The goal of a nursing diagnosis is to identify health problems of the patient and his/her family. A diagnosis provides a direction for the following nursing care [2]. The nurses check each patient's conditions based on his/her observations against the definitions and the defining characteristics of various diagnoses in the hierarchically organized nursing diagnoses ontology to select the most appropriate diagnoses. This step will result in a number of potentially relevant nursing diagnoses. The nurses will verify each diagnosis and then select one or more diagnoses as the final outcome.

2 Collaborative Nursing Care Plan Generation

For the nurses to generate a quality nursing care plan, the nurses need to follow a number of steps starting from the data collection stage and ending with the nursing diagnoses, outcome, and intervention selection stage. The most difficult stage of the nursing care plan generation is the selection of the most appropriate nursing diagnosis given the physical and mental state and symptoms of a patient.

There are 167 nursing diagnoses, which are organized as an ontology of a three-layer hierarchy. At the top, there are 13 domains. The domains are: health promotion; nutrition; elimination; activity/rest; perception/cognition; self-perception; role relationship; sexuality; coping/stress tolerance; life principles; safety/protection; comfort; and growth/development [3]. In the middle layer, there are 46 classes. For example, five classes under the 'nutrition' domain are: ingestion, digestion, absorption, metabolism, and hydration. The nursing diagnoses occupy the bottom layer.

Each diagnosis is associated with the definition and one or more defining characteristics. The nurses learn to navigate the hierarchy and differentiate diagnoses by reviewing the existing cases and also by observing what other nurses do. Nursing

outcomes are also organized as an ontology of a multi-level hierarchy. In addition, each diagnosis is linked to one or more potentially suitable nursing outcomes. Therefore, the nurses select the most appropriate nursing outcomes based on the nursing diagnosis, the state, and the symptoms of the patients. Similarly, the nursing interventions are organized as an ontology of a multi-level hierarchy. Each diagnosis and outcome is linked to one or more potentially suitable nursing interventions. This leads to the selection of the most appropriate nursing interventions based on the nursing diagnosis, nursing outcome, the state, and the symptoms of the patients. Yet again, the nurses learn to select the correct outcomes and interventions by reviewing the existing cases and also by observing what other nurses do namely exploring the outcomes and interventions ontology.

Many nurses are responsible for many patients. The patients come and go. The nurses in many critical settings work in the three eight-hour shifts. Therefore, it is essential and necessary for many nurses to share the generation and the use of the nursing care plan. One nurse might not be able to collect all necessary information about a patient to reach a diagnosis at one time. Thus, other nurses will need to collaborate in the data collection stage. In addition, one nurse might develop doubts, questions, or further data collection needs while trying to select the most appropriate nursing diagnosis. Other nurses can either find then share answers to the questions for others including the nurse to study or complete the nursing diagnosis step with the additional information.

3 Collaborative Learning System

We are developing a collaborative nursing care plan generation system for the nursing science undergraduate students. Every semester, the Juniors, who major in nursing science at the Department of Nursing Science, Konkuk University in Chungju, Korea, develop detailed care study reports of patients while they work as a student intern at a psychiatric warden for four weeks. The nursing care plan is one section of the case report, which is submitted at the end of the internship period. Traditionally, each case study report including the nursing care plan was prepared by one student. However, we wanted to introduce the group project aspect into the internship assignment by having the students to generate the nursing care plan collaboratively.

To enable the easy collaboration amongst the group members, we developed a web community for the group members to communicate and also to collaboratively generate the nursing care plan. The group members can create a discussion thread at will to start an online discussion about a new subject or participate in an existing discussion thread to argue for or against a posted opinion. A discussion thread can be also used as a repository of the collected data.

For the web-based collaborative nursing care plan generation, we made each section of the care plan as a separate discussion board within the web community. The students can post the nursing care plan contents as the messages for others to review and comment. After everyone accepts the posting, it becomes a part of a text repository, which eventually becomes the nursing care plan.

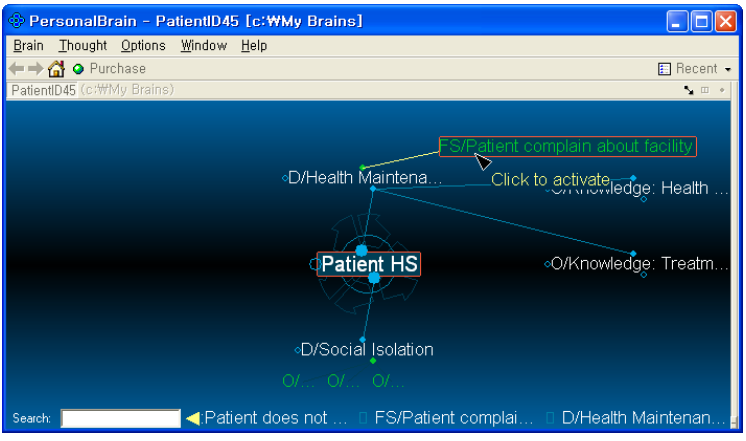


Fig. 1. Mind Map for Patient HS

We learned that the students selected one or more of the appropriate diagnoses by exploring the nursing diagnoses ontology from the top layer while constantly referring to the collected factual information about the patients. The majority of the students were using the breadth first search strategy [4].

To help the students learn the structure and the contents of each ontology, we incorporated an interactive visualization software component to allow visual inspection of the ontology and the collected factual information about the patient. The students can link one or more diagnoses to a patient using the visualizer. The representation is inspired by MindMap, which is an alternative way to organize the patient information especially devised to enhance the educational experiences by the nursing students [5]. The Figure 1 shows a mind map representation about a patient, who is referred as HS. We used PersonalBrain, a commercial off the shelf software from The Brain Technologies Corp (http://www.thebrain.com/). PersonalBrain is an information visualization system, which enables the users to link information into a network of logical associations. The Figure 1 shows 'Health Maintenance, Altered' and 'Social Isolation' as the nursing diagnoses for the patient HS. Each nursing diagnosis is preceded by 'D'. Nursing Outcome is preceded by 'O' and Intervention is preceded by 'I'. PersonalBrain truncates the labels of each node in the graph if the label is too long. The full label is revealed when the user moves the cursor on top of the node. The user can make any node a central node by double clicking on it. The mind map represented in PersonalBrain is dynamically repositioned to minimize the overloading of the visual representation in small screen space. At the top of the mind map, there is a factual information summary observed by the nurse. "Patient complains about facility' is preceded by 'FS', which stands for 'factual summary'. 'E' is for the evaluative statement by the nurses.

4 Conclusion

We are in the early stages of developing a collaborative nursing care plan generation system where understanding the contents and the structure of the nursing diagnoses

ontology, outcomes ontology, and intervention ontology by the students is essential. We developed a prototype system, which enables the students to inspect the nursing diagnoses onotology to aid in fining the most appropriate diagnoses. We are in the process of the evaluating the system. We expect the resulting system to aid both nursing students and practitioners. They can learn from each other by working toward the same goal of generating a quality nursing care plan.

References

1. Doenges, M., and Moorehead, M.F., Application of Nursing Process and Nursing Diagnosis: An Interactive Text for Diagnostic Reasoning 4th Edition, F.A. Davis Co., Philadelphia, 2003.
2. Sparks. S.M. and Taylor, C.M., Nursing Diagnosis Reference Manual 5th Edition, Springhouse, Springhouse, PA, 2000.
3. Ralph, S.S., Craft-Rosenberg, M., Herdman, T.H., and Lavin, M.A., NANDA Nursing Diagnoses: Definitions & Classification 2003-2004, NANDA International, Philadelphia, 2003.
4. Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C., Introduction to Algorithms 2nd Edition, McGraw-Hill, Boston, 2001.
5. Mueller, A., Johnston, M. and Bligh, D., Joining mind mapping and care planning to enhance student critical thinking & achieve holistic nursing care. Nursing Diagnosis 13(1):24-27, 2002.

Generating and Evaluating Triples for Modelling a Virtual Environment

Marie-Laure Reinberger[1] and Peter Spyns[2]

[1] CNTS - University of Antwerp,
Universiteitsplein 1, B-2610 Wilrijk, Belgium,
marielaure.reinberger@ua.ac.be
[2] STAR Lab - Vrije Universiteit Brussel,
Pleinlaan 2 Gebouw G-10, B-1050 Brussel, Belgium
Peter.Spyns@vub.ac.be

Abstract. Our purpose is to extract RDF-style triples from text corpora in an unsupervised way and use them as preprocessed material for the construction of ontologies from scratch. We have worked on a corpus taken from Internet websites and describing the megalithic ruin of Stonehenge. Using a shallow parser, we select functional relations, such as the syntactic structure subject-verb-object. The selection is done using prepositional structures and frequency measures in order to select the most relevant triples. Therefore, the paper stresses the choice of patterns and the filtering carried out in order to discard automatically all irrelevant structures. At the same occasion, we are experimenting with a method to objectively evaluate the material generated automatically.

1 Introduction and Background

The development of ontology-driven applications is currently slowed down due to the knowledge acquisition bottleneck. Therefore, techniques applied in computational linguistics and information extraction (in particular machine learning) are used to create or grow ontologies in a period as limited as possible with a quality as high as possible.

This paper wants to report on experiments concerning the automatic extraction of semantic information from text. Therefore, the results of shallow parsing techniques are combined with unsupervised learning methods. The results we are reporting here have been evaluated automatically, and the final purpose of this study is to establish an ontology-based method for designing virtual web environments, e.g., a graphical representation of Stonehenge. To this aim, we build a repository of lexical semantic information from text, ensuring evolvability and adaptability. This repository can be considered as a complex semantic network. Currently, the focus is on the discovery of concepts and their conceptual relationships, although the ultimate aim is to discover semantic constraints as well. We have opted for extraction techniques based on unsupervised learning methods [14,15,18,16] since these do not require specific external domain knowledge such as thesauri and/or tagged corpora [1].

The following section (2) presents the unsupervised text miner, its theoretical foundations and its working principles. Subsequently, in section 3, we discuss the metrics

[1] Except the training corpus for the general purpose shallow parser.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1205–1214, 2005.
© Springer-Verlag Berlin Heidelberg 2005

with which we evaluate the mining results, which are presented in (section 4). Calculated scores are provided in section 5. An overall discussion (section 6) precedes the conclusion and a future work section (7) that ends this paper.

2 Unsupervised Mining

The ***linguistic assumptions*** underlying this approach are:

1. the principle of selectional restrictions (syntactic structures provide relevant information about semantic content), and
2. the notion of co-composition [13] (if two elements are composed into an expression, each of them impose semantic constraints on the other).

The fact that heads of phrases with a subject relation to the same verb share a semantic feature would be an application of the principle of *selectional restrictions*. The fact that the heads of phrases in a subject or object relation with a verb constrain that verb and vice versa would be an illustration of *co-composition*. In other words, each word in a noun-verb relation participates in building the meaning of the other word in this context [7]. If we consider the expression "write a book" for example, it appears that the verb "to write" triggers the informative feature of "book", more than its physical feature. We make use of both principles in our use of clustering to extract semantic knowledge from syntactically analysed corpora.

In a specific domain, an important quantity of semantic information is carried by the nouns. At the same time, the noun-verb relations provide relevant information about the nouns, due to the semantic restrictions they impose. In order to extract this information automatically from our corpus, we used *memory-based learning* [6,9] and more specifically the ***memory-based shallow parser*** which is being developed at CNTS Antwerp and ILK Tilburg [2,5] [2]. This shallow parser takes plain text as input, performs tokenisation, POS tagging, phrase boundary detection, and finally finds grammatical relations such as subject-verb and object-verb relations (tagged as NP-SBJ and NP-OBJ in the example below, a number indicating the subject, the verb and the object that belong to the same proposition), which are particularly useful for us. The software was developed to be efficient and robust enough to allow shallow parsing of large amounts of text from various domains .

Example: *[NP-SBJ-1 The/DT Sarsen/NNP lintels/NNS NP-SBJ-1] [VP-1 form/VBP VP-1] [NP-OBJ-1 a/DT continuous/JJ circle/NN NP-OBJ-1] PNP [PP around/IN PP] [NP the/DT top/NN NP] PNP ./.*

Different methods can be used for the ***extraction of semantic information*** from parsed text. Pattern matching [1] has proved to be an efficient way to extract semantic relations, but one drawback is that it involves the predefined choice of the semantic relations that will be extracted. On the other hand, clustering [10,12] only requires a minimal amount of "manual semantic pre-processing" by the user but an important amount of data. But both allow for an unsupervised process [3,4]. Here, as we consider

[2] See `http://ilk.kub.nl` for a demo version.

a small corpus (4k words), we have employed pattern matching. We get results using pattern matching methods on syntactic contexts to also extract previously unexpected relations.

The initial *corpus* is formed of descriptions of Stonehenge collected from websites. But as some texts were providing historical information about the building of Stonehenge over the centuries and the modifications that took place in the arrangement of stones over the years, some of them disappearing or being moved, we had to adapt the existing descriptions to keep only information relating to what Stonehenge looked like at a certain moment of its history. Therefore, we had to discard some sentences and the Stonehenge we aim to represent is the arrangement referred to by historians as the fourth age of Stonehenge. At the same time, it appeared that the material available on the Internet was not complete enough and was lacking practical information concerning the disposition of the stones, as that information was provided by pictures displayed on the websites. Therefore, we have completed the corpus with literal descriptions based on those sketches. Those descriptions represent less than a tenth of the final corpus.

As mentioned above, the shallow parser detects the subject-verb-object *structures*, which gives us the possibility to focus in a first step on the term-verb-term relations with the terms appearing as the noun phrase (NP) of the subject and object phrases (NP_Subj and NP_Obj). This type of structure features a functional relation between the verb and the term appearing in object position. We call this functional relation a lexon. The pattern we are using is checking also the existence of a prepositional structure (Prep and NP) following the direct object NP:

NP_Subj-V-NP_Obj[-Prep-NP]

As intransitive verbal constructions contain also a lot of functional information, we rely on the chunks output to extract the verbal structures involving intransitive verbal structures:

NP-V-Prep-NP[-Prep-NP]

By using both those patterns, we retrieve a set of functional relations that we have now to sort out, as some of them contain mistakes due to errors at the syntactic level or during the relation extraction process.

Therefore, in a next step, we create a new set of relations, using this time information contained in prepositional structures. Those patterns are more frequent in the corpus and easier to detect. The prepositional patterns are defined using word frequencies and a predefined prepositions set focusing on spatial relations determining dimensions, distances and positions: in, of, within, inside, into, around... The general pattern used with respect to the chosen preposition is:

NP-Prep-NP[-Prep-NP]

Some prepositional structures are then selected using a frequency measure. Only the structures including one of the N most frequent nouns are kept. In order to perform this ranking, the NPs are lemmatized.

The last selection process consists in using the prepositional structures selected to filter out incorrect functional relations from the previous set. We select the lexons that contain at least one NP appearing in a prepositional pattern.

Eventually, *a last filtering operation* consists in comparing the strings two by two and always keeping the longest one, therefore the most complete. E.g., the sentence: "The bluestones are arranged into a horseshoe shape inside the trilithon horseshoe." will produce two lexons (1) and (2), of which the former one will be discarded and the latter one kept.

(1) bluestones arranged into horseshoe shape
(2) bluestones arranged into horseshoe shape inside trilithon horseshoe.

3 Evaluation Metrics

3.1 Term Relevance

As the three constituting elements of a triple or lexon resulting from the mining consist of words appearing in the domain corpus, we can investigate to what extent the vocabulary of the triples adequately represents the notions of a particular application domain [20]. In the following sections, some metrics to measure the adequacy between the domain vocabulary and the triples vocabulary are presented. See [18] for details on the metrics.

Coverage. A simplistic metric to determine the coverage would be to calculate the intersection between the vocabulary of the triples and the entire corpus. As many words do not represent domain concepts (e.g. adverbs, determiners, particles, ..., which are by definition not retained by the unsupervised text miner) the triples generated automatically most probably will not attain a high domain coverage rate. In order to differentiate more important words from less important ones, the frequency of a word can be taken into account. Naively, one would expect that important domain words are mentioned more often than others. Therefore, the words are grouped into frequency classes, i.e. the absolute number of times a word appears in a corpus.

coverage(triples,text) =

$$\frac{\sum_{i=1}^{n} \frac{\#(words_triples_freq_class_i \bigcap words_text_freq_class_i)}{\#words_text_freq_class_i} * 100}{n}$$

Precision and Recall. In the approach proposed here, we use a metric from quantitative linguistics to automatically build a gold standard. The standard consists of a set of words that characterise an application domain text and result from a quantitative comparison with another text. Regarding technical texts, one can easily assume that the specialised vocabulary constitutes the bulk of the characteristic vocabulary, especially if the other corpus with which to compare is the Wall Street Journal (= collection of general newspaper articles), as is the case here. A word is said to be statistically relevant or not, with a 95% confidence level, based on computed z-values expressing the relative difference between two proportions, i.e., the word frequencies in a technical text (the Privacy Directive) vs. in a general text (WSJ) [8].

recall(triples,text) =

$$(\frac{\#(words_of_triples_mined \bigcap statistically_relevant_words)}{\#statistically_relevant_words}) * 100)$$

precision(triples,text) =

$$(\frac{\#(words_of_triples_mined \bigcap statistically_relevant_words)}{\#words_of_triples_mined}) * 100)$$

Accuracy. The purpose of calculating the accuracy is to refine the coverage measure that is based only on word frequency, by combining it with the precision measure. The source of inspiration is Zipf's law [22]. Stated simply, in each text there is a small set of words that occur very often and a large set of words that rarely occur. Zipf has discovered experimentally that the more frequently a word is used, the less meaning it carries (e.g., stop words). A corollary from Zipf's law is that domain or topic specific vocabulary is to be looked for in the middle to lower frequency classes. Consequently, triples mined from a corpus should preferably contain terms from these "relevant" frequency classes. Luhn [11] has defined intuitively a frequency class upper and lower bound. The most significant words are found in the middle of the area of the frequency classes between these boundaries. He called this the "resolving power of significant words".

Here, the frequency classes that contain a high number of statistically relevant words will be considered as "relevant" frequency classes. Currently, we assume that a frequency class should contain at least 60% of relevant words in order to be a relevant class. Remark that accuracy is the coverage but applied to relevant frequency classes only.

accuracy(triples,text) =

$$\frac{\sum_{i=1}^{n} \frac{\#(words_triples_rel_freq_class_i \bigcap words_text_rel_freq_class_i)}{\#words_text_rel_freq_class_i} * 100}{n}.$$

3.2 Triple Relevance

After having determined how well (or bad) the overall triple vocabulary covers the terms representing important notions of the domain, individual triples in their entirety can be examined. Again, the relevant lemmas as used as reference. A triple is considered relevant if composed by at least two relevant constituents. We did not use a stopword list, as this list might change with the nature of the corpus, and as a preposition can be potentially relevant (unlike e.g. in information extraction applications) since they are included in the lexons automatically generated. The metrics should be able to cope with these issues. For this experiment, the text miner produced triples consisting of a noun phrase in the subject and object positions and a verb or preposition in the predicate position. If a relevant term is included in the noun phrase or matches with the verb or preposition, that part of the triple is considered as relevant.

4 Results

Our extraction process results in different kinds of relations, and on one hand we retrieve a small amount of relations that refer to world knowledge, such as *bottom of stone, shape of stone, centre of circle*. But our main interest lies in more specific spatial

relations and more generally information related to the disposition of the stones, the shapes of the different arrangements of stones, as well as the positions of the different stone arrangements, one in respect to the other. At the same time, some more general relations like 1. and 2., can allow us to check or confirm more precise ones, or just acknowledge the existence of an element of the monument.

1. ring of bluestones
2. central part of monument
3. monument comprises of several concentric stone arrangement
4. Bluestone circle outside Trilithon horseshoe
5. Bluestone circle inside Sarsen Circle
6. Bluestone circle is added outside Trilithon horseshoe
7. 100 foot diameter circle of 30 sarsen stone

We give below some examples of bad relations we are extracting. Those relations are either incomplete, irrelevant or unfocused.

- Altar Stone is in front
- Sarsen block are 1.4 metre
- 120 foot from ring

Those erroneous relations are due to long or complex sentences on which mistakes happen. Those errors can take place during the syntactic analysis, because of the bad tagging of a word that is unknown to the parser (such as “trilithon” for example) and will lead to a wrong analysis of the whole chunk. They can also take place during the pattern matching, if the syntax of the sentence is unusual and does not fit the definition of the pattern we are using. We extract also a lot of correct relations that we did not use as they were not relevant for the purpose of representing Stonehenge graphically.

- Aubrey Holes vary from 2 to 4 foot in depth
- bluestone came from Preselus Mountain in southwestern Wale
- carving on twelve stone

Those relations provide us with information concerning the provenance of the stones, the location of Stonehenge itself, the sizes and weights of the stones, as well as some information describing carvings on stones. But we need to know how many of those relations are really useful, and how well we can hope to perform in building a virtual Stonehenge, only using those relations.

5 Evaluation

5.1 Term Relevance

We stress that text mining does not deal with an actual conceptualisation, but rather with its representation or lexicalisation in a text, meaning that we cannot access directly the conceptualisation (meaning level) [19]. Therefore, the triples generated automatically are in fact more a kind of first stage processed material. For the moment, hierarchical

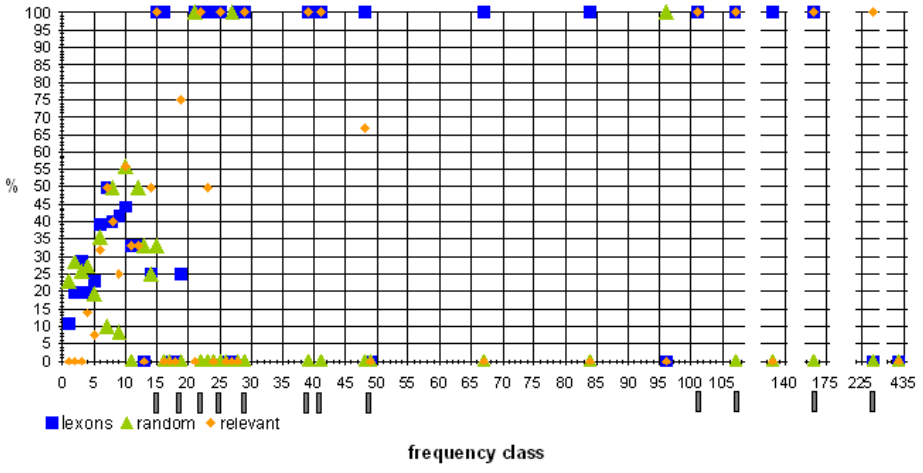


Fig. 1. Coverage per frequency class for the lexon lemmas, randomly selected and statistically relevant lemmas

and other relationships are not yet assessed. The quasi totality of the sentences of the corpus provided at least one triple. The sentences containing a verbal structure that proves too complex to be handled by the shallow parser in most cases provided triples based on a prepositional structure.

A baseline (randomly selected terms) has been calculated to illustrate that both the statistical formula and the unsupervised mining techniques clearly do not correspond to a random selection (see the triangles in Figure 1). The squares show how many lemmas in a FC belong to the lexons or triples vocabulary, while the diamonds indicate how many words of a FC are statistically considered as relevant. The same figure shows that additional lower frequency words should be retained but the accuracy (relevant frequency classes are marked by a little bar box below the frequency class rank) seems to be rather high (85,41%). The overall coverage is 54,49% while the overall recall goes up to 84,12% with a precision of only 26,38%. Note that there are many one-member classes resulting in 0% or 100 % scores, especially from FC 28 onwards.

5.2 Triple Relevance

On a total of 204 lexons or triples generated by the unsupervised text miner, 177 triples are considered as relevant (110 triples containing a relevant word at all three positions). Table 2 displays some triples that have been rejected by the evaluation procedure. It is indicated whether the subject (S), predicate (P) or object (O) element of a triple is considered as relevant ('+') or not ('-'). Some examples are shown in Table 1.

5.3 Qualitative Evaluation

Another evaluation of the relations has been performed manually, by humans. Although the humans had a vague notion of what Stonehenge was, they were initially unable to

Table 1. Some triples generated automatically and evaluated as relevant (95% confidence)

SPO	triple
-++	<base, be, from ditch>
+++	<bluestone oval, become, horseshoe shape>
+-+	<circular bank, with, ditch>
+++	<least bluestone, stand of, stone>
+-+	<large sarsen stone, laid, shape>
+++	<outer sarsen circle, be, horseshoe >
+++	<trilithon, lintel of, large sarsen stone>
+++	<wooden structure, be build, inside circle>

Table 2. Some triples generated automatically but rejected (95% confidence)

triple
<area, fire as, range >
<bluestone circle, with, famous >
<carving, on, stone >
<long processionway, with, bank ditch >
<heel stone, lean, inward>
<sarsen trilithon, have, opening>
<stone, add, evidence>
<sparkling 8-ton bluestone, go, heart focus >

draw it. Subsequently, two drawings have been elaborated using respectively only the corpus and only the relations. These have been compared. A complete report can be found in [17]; here, we will simply summarise the most important findings.

Overall, the differences between the two drawings show that the set of relations automatically extracted contains exhaustive enough and relevant enough information concerning the amount of stones and their disposition to draw a clearly recognisable Stonehenge. The relations are precise enough to determine the general structure and all the main stone arrangements with correct relative positions. The orientation of the monument, with its opening to the North East, is present also. We are principally missing information concerning distances between the different structures and some specific and less prominent elements of the arrangement. In a nutshell, the relations we are extracting automatically allow the initiation of a graphical representation of Stonehenge, but they need enrichment for a complete and accurate drawing of the monument.

6 Discussion

Our automatic extraction process allows us to retrieve information related to the positions of stones, their amounts, sizes, weights, as well as their composition or their shape. Very often, a relation can be double checked due to the fact that the corpus is composed of the content of various documents concatenated. They provide different descriptions of Stonehenge containing of course some common information. In some cases, a similar information can be retrieved in a same sentence with two different patterns.

We mainly miss information concerning distances, and specific elements such as the avenue or the rings of pits. The principal reason for this phenomenon is the length of some sentences resulting in a wrong syntactic analysis or in a partial detection of the pattern. In other cases, the information is not appearing in every description, and therefore is not rated high enough and considered as not relevant by the system to be kept. One solution would be to use more patterns, and to define them more precisely, but we want to keep on working in an unsupervised way and rely on the amount of relations extracted and their quality. An important amount of world knowledge is lacking, as this information is not provided in descriptions aimed at readers who know what a stone might look like and that it will tend to lie on the ground.

The mined results that have passed the automatic evaluation demonstrate that the miner produces precise results (86,76%). Errors seem to be mainly due to a tagging error: a noun is taken for a verb. The human evaluators were able to draw the basic structures of Stonehenge without any difference to a prior experiment that did not use the automatic evaluation procedure (for more details see [17]). More work on assessing the practical usability of the results and the corresponding evaluation remains to be done.

7 Conclusion and Future Work

Regarding the representation of Stonehenge and the manual evaluation which are described above, we would have needed more material, and especially more accurate textual descriptions of the site. This would certainly improve the quality of the relations and increase the amount of information they convey. All necessary information is not (always) available on the Internet. Visual representations require precise and exhaustive descriptions that are not aimed at human beings. For this reason, collecting enough relevant material proved not to be an easy task in that particular domain. The automatic evaluation shows that too much garbage is generated, but that the relevant frequency class "threshold" results in discarding quite some noise. The manual evaluation allows us to conclude that the triples generated by the unsupervised miner are helpful to a human knowledge engineer. The exact extent of usefulness requires further investigation.

We are also planning to try to improve the syntactic analysis by training the shallow parser on a corpus containing information such as semantic relations. We are expecting a better output, especially concerning the detection of subjects and objects, as well as the possibility to detect more specific structures such as locations or temporal complements.

By using the automated evaluation metrics, it will be possible to set up regression tests to monitor and quantify the potential performance improvement of the miner.

Acknowledgements. This research has been carried out during the OntoBasis project (IWT GBOU 2001 #10069), sponsored by the IWT Vlaanderen (Institution for the Promotion of Innovation by Science and Technology in Flanders).

References

1. Matthew Berland and Eugene Charniak. Finding parts in very large corpora. In *Proceedings ACL-99*, 1999.
2. Sabine Buchholz. Memory-based grammatical relation finding. In *Proceedings of the Joint SIGDAT Conference EMNLP/VLC*, 2002.
3. Sharon A. Caraballo. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings ACL-99*, 1999.
4. Philipp Cimiano, Steffen Staab, and Julien Tane. Automatic acquisition of taxonomies from text: FCA meets NLP. In N. Nicolov, R. Mitkov, G. Angelova, K. Boncheva (eds.) *Proceedings of the ECML/PKDD Workshop on Adaptive Text Extraction and Mining ATEM03*, pages 10–17, 2003.
5. Walter Daelemans, Sabine Buchholz, and Jorn Veenstra. Memory-based shallow parsing. In *Proceedings of CoNLL-99*, 1999.

6. Walter Daelemans, and Antal Van den Bosch. *Memory-Based Language Processing*. Cambridge University Press, 2005.
7. Pablo Gamallo, Alexandre Agustini, and Gabriel P. Lopes. Using co-composition for acquiring syntactic and semantic subcategorisation. In *Proceedings of the Workshop SIGLEX-02 (ACL-02)*, 2002.
8. Josse De Kock. *Elementos para una estilística computacional - tomo*. Editorial Coloquio, Madrid, 1984.
9. Sandra Kuebler. Parsing Without Grammar – Using Complete Trees Instead. In N. Nicolov, R. Mitkov, G. Angelova, K. Boncheva (eds.) *Recent Advances in Natural Language Processing III: Selected Papers from RANLP 2003*. Amsterdam: John Benjamins, 2004.
10. Dekang Lin. Automatic retrieval and clustering of similar words. In *Proceedings of COLING-ACL-98*, 1998.
11. H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159 – 195, 1958.
12. Patrick Pantel and Dekang Lin. Discovering word senses from text. In *Proceedings of ACM SIGKDD-02*, 2002.
13. James Pustejovsky. *The Generative Lexicon*. MIT Press, 1995.
14. Marie-Laure Reinberger, Peter Spyns, Walter Daelemans, and Robert Meersman. Mining for lexons: Applying unsupervised learning methods to create ontology bases. In Robert Meersman, Zahir Tari, and Douglas Schmidt et al., editors, *On the Move to Meaningful Internet Systems 2003: CoopIS, DOA and ODBASE*, volume 2888 of *LNCS*, pages 803 – 819, Berlin Heidelberg, 2003. Springer Verlag.
15. Marie-Laure Reinberger, Peter Spyns, A. Johannes Pretorius, and Walter Daelemans. Automatic initiation of an ontology. In Robert Meersman and Zahir Tari et al., editors, *On the Move to Meaningful Internet Systems 2004: CooPIS, DOA and ODBASE (part I)*, volume 3290 of *LNCS*, pages 600 – 617. Springer Verlag, 2004.
16. Marie-Laure Reinberger and Peter Spyns. *Ontology Learning from Text: Methods, Applications and Evaluation*, chapter Unsupervised Text Mining for the Learning of DOGMA-inspired Ontologies. IOS Press, Amsterdam, 2005.
17. Marie-Laure Reinberger. Automatic extraction of spatial relations. In *Proceedings of the TEMA workshop, EPIA 2005, Portugal*
18. P. Spyns and M.-L. Reinberger. Lexically evaluating ontology triples automatically generated from text. In A. Gómez-Pérez and J. Euzenat, editors, *Proceedings of the second European Semantic Web Conference*, volume 3532 of *LNCS*, pages 563 – 577. Springer Verlag, 2005.
19. Peter Spyns and Jan De Bo. Ontologies: a revamped cross-disciplinary buzzword or a truly promising interdisciplinary research topic? *Linguistica Antverpiensia, new series*, (3), 2004.
20. York Sure, Gomez-Perez Gomez-Perez, Walter Daelemans, Marie-Laure Reinberger, Nicola Guarino, and Natalya F. Noy. *Why Evaluate Ontology Technologies? Because It Works!*, In IEEE Intelligent Systems, vol. 19, no. 4, pp. 74-81, July/August 2004.
21. C. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979.
22. George K. Zipf. *Human Behaviour and the Principle of Least-Effort*. Addison-Wesley, Cambridge MA, 1949.

An Ontology-Driven Approach for Modeling Behavior in Virtual Environments

Bram Pellens, Olga De Troyer, Wesley Bille**,
Frederic Kleinermann, and Raul Romero

Research Group WISE, Vrije Universiteit Brussel,
Pleinlaan 2 - 1050 Brussel, Belgium
Bram.Pellens@vub.ac.be
http://wise.vub.ac.be

Abstract. Usually, ontologies are used to solve terminology problems or to allow automatic processing of information. They are also used to improve the development of software. One promising application area for ontologies is Virtual Reality (VR). Developing a VR application is very time consuming and requires skilled people. Introducing ontologies in the development process can eliminate these barriers. We have developed an approach, VR-WISE, which uses ontologies for describing a Virtual Environment (VE) at a conceptual level. In this paper we will describe the Behavior Ontology, which defines the modeling concepts for object behavior. Such an ontology has several advantages. It improves the intuitiveness; facilitates cross-platform VR development; smoothens integration with other ontologies; enhances the interoperability of VR applications; and allows for more intelligent systems.

1 Introduction

As a result of better software systems in conjunction with cheaper and faster hardware systems, Virtual Environments (VEs) are used in many different fields nowadays. Also, the availability of the Web together with the development of 3D formats specific for the Web makes Virtual Reality (VR) technology accessible to a broader audience. However, the development of a VR application (even for the Web) is still a difficult and specialized activity and is usually done by a VR-expert. One of the reasons for this is the fact that the development of a VR application is mainly an implementation activity. If we compare the development process of a VR application with the development of classical software, we see that the design phase in the development process of a VR application (from the perspective of a classical software engineering life cycle) is usually a very informal activity on which little time is spent. Few formal techniques in the context of VR exist to support this phase effectively. A systematic approach that uses the output of the design phase as input for the implementation phase does not exist. Especially for the behavioral part of a VR application no true design is done.

** Research Assistant for the Fund for Scientific Research - Flanders.

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 1215–1224, 2005.
© Springer-Verlag Berlin Heidelberg 2005

We believe that the introduction of ontologies in the design process will facilitate the development of a VE. The use of ontologies in the development process has several advantages: (1) the VE can be specified at a conceptual level, which will not only provide a more intuitive way to specify a VE but will also facilitate cross-platform VR development; (2) smooth integration with other ontologies (like domain ontologies) is possible; (3) because semantic descriptions are available it will enhance the interoperability of VR applications; (4) the use of ontology technology allows for more intelligent systems (ontology technology provides reasoning mechanisms). Therefore, we developed an ontology-based approach called VR-WISE [3] that provides an explicit conceptual design phase for VR applications. A set of high-level modeling concepts is provided to allow modeling a VR application using knowledge from the application domain.

In this paper we describe how VR-WISE employs ontologies in its approach and in particular we will describe the ontology that provides the modeling concepts for object behavior. This ontology allows specifying the behaviors following an action-oriented paradigm; meaning that the focus is on the actions that an object needs to perform rather then on the states the object can be in. Furthermore, the behavior can be specified separated from the definition of the objects itself and from the interactions used to invoke the behavior.

The paper is organized as follows. Section 2 discusses some related work. In section 3, we introduce the VR-WISE approach and the different types of ontologies involved. Section 4 describes the behavior ontology, which is used within our approach for modeling the behavior in a VE system. The software tool that has been implemented to support this research is briefly discussed in section 5. We end this paper with a conclusion and future work in section 6.

2 Related Work

Developing VEs in general and modeling object behavior in particular has never been easy or intuitive for non VR-skilled persons. A number of research groups have developed ways to facilitate the specification of VEs using ontologies.

In [8] the idea of Semantic Virtual Environments is proposed. The semantics of a VE is exposed by means of RDF descriptions in order to cope with the heterogeneity in data models and network protocols. Both descriptions, the contents of the VE as well as the semantic information need to be created separately. However, as we will see in the following section, using the VR-WISE approach and the associated ontologies, we automatically obtain the semantic description. There is no need to annotate the VE afterwards.

The use and the advantages of ontologies for discrete-event simulation and modeling are investigated in [7]. The DeMO ontology is an OWL based ontology that has been created for this purpose. The ontology tries to give a classification of concepts used in discrete event models. This approach focuses more on the different types of models their concept inter-relationships while the VR-WISE approach focuses on the intuitiveness of the concepts and its usability for novice users.

The STEP language is a scripting language that is mainly designed for the specification of communicative acts on embodied agents in the VE [5]. The STEP ontology defines all the concepts and relationships related to human animation. However, the ontology is only directed towards a specific domain and is therefore not usable for general VR application development. The VR-WISE approach provides a set of modeling concepts generally applicable to any VE.

Another system that is similar to ours is WordsEye which can be used for automatically converting text descriptions into 3D representations [2]. Here, a high-level representation, as the outcome of linguistic analysis from text, is translated into low-level depictors to be visualized. VR-WISE uses ontologies for describing the VE which allows to infer additional information during the modeling process and allows for searching in and reasoning with the VE afterwards.

3 VR-WISE Approach

We first give a general overview of our approach to develop VR applications using ontologies. The main goal of the research is to facilitate and shorten the development process of VR applications by means of conceptual specifications (also called *conceptual models*). A conceptual specification is a high-level representation of the objects in the VE, how they are related to each other, how they will behave and interact with each other and with the user. Such a conceptual specification must be free from any implementation details and not influenced by the current technical limitations of the VR technology. The use of a conceptual model may also improve the reusability, extensibility and modularity of the VR application.

In VR-WISE, ontologies [3] play an important role. Ontologies are used for two different purposes. (1) Ontologies are used explicitly during the design process for representing knowledge about the domain under consideration. (2) Ontologies are also used as general information representation formalism (internally). As a consequence, different types of ontologies are used: *domain ontologies*, *application ontologies* and *modeling ontologies*. Domain ontologies are used for the description of the objects from the relevant domain that should be used in the VE. Application ontologies are used to capture the properties of the VE itself. The modeling ontologies are used to capture the modeling concepts provided as well as general information about VR implementation environments. We will describe their purpose and their role while explaining the approach.

The development process in the VR-WISE approach is divided into three (mainly) sequential steps (see figure 1).

Specification Step. The specification step covers the design phase. It allows the designer to specify the VE at a high level using domain knowledge and without taking into account any implementation details. The specification is done at two levels: a type level and an instance level. The type level is specified using a *Domain Ontology* and describes the concepts needed from the domain under consideration. The Domain Ontology describes the domain concepts by means of their properties as well as their relationships. For example, in the architectural

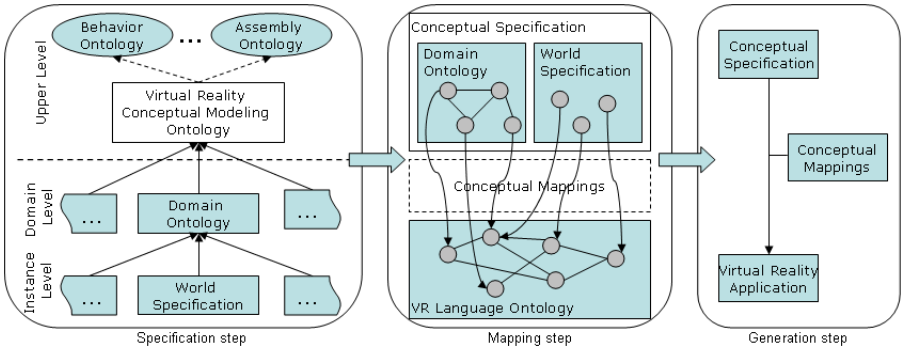


Fig. 1. Overview of the VR-WISE Approach

domain, this ontology would contain concepts like Wall, Door, Window, Beam, and relationships such as "a Door is always located in a Wall", "a Room consists of a number of Walls". It is possible that such a domain ontology already exists (created for other purposes). In that situation, this ontology can be reused.

The instance level is specified by means of the *World Specification*, which will contain the actual conceptual description of the VE to be built. In fact, it describes the population of the VE. This World Specification is created by instantiating concepts from the Domain Ontology. These instances represent the objects that will populate the VE. For the architectural example, there will be a number of Wall-instances, multiple Window-instances and Door-instances. In addition, instance-specific information (e.g. values for properties like size, color; location and orientation) and information specific for the world itself (e.g. gravity, lights, ...) is given in the World Specification.

To define the concepts, their properties and relationships, a number of high-level modeling concepts are provided. These modeling concepts are independent of any application domain and are defined in a modeling ontology, namely the *Virtual Reality Conceptual Modeling Ontology*. In fact, it is a collection of different ontologies. Modeling a VE involves many different aspects: the objects in the world, their properties and composition, their behavior, the interaction between objects and the interaction with the user, etc. In this paper we will focus on the *Behavior Ontology* that defines the concepts for specifying object behavior.

Mapping Step. In the mapping step, the mapping from the conceptual level to the implementation level is specified. As this step is not important for the purpose of this paper, we will not go into details. The purpose of this step is defining the mappings from the concepts in the Domain Ontology and World Specification to VR implementation primitives. The low-level VR concepts that can be used as target in the mappings are described in an ontology called the *Virtual Reality Language Ontology*. This ontology defines concepts that are commonly available in VR implementation environments.

Generation Step. The generation step generates the actual source code for the VE specified in the specification step using the mappings defined in the

mapping step, i.e. a working application is automatically generated. More details about the approach can be found in [1]. In summary, we can state that the approach provides a systematic way to make a conceptual design for a VE and to convert it into a working VR application. The ontologies that are built during the specification phase provide a semantic description of the VE that can be used for many different purposes.

4 The Behavior Ontology

In the previous section, the general approach of VR-WISE was introduced. We explained that different types of ontologies are used. One of these is the VR Conceptual Modeling Ontology that defines all modeling concepts provided by VR-WISE. This ontology can be characterized as an upper ontology since it is acting as a library of domain independent concepts. In this section we will elaborate on one of its sub-ontologies, the Behavior Ontology. The purpose of this section is twofold. On the one hand we will show how it is possible to specify behavior of objects at a conceptual level, and on the other hand we show how the modeling concepts used for this can be defined by means of an ontology. The Behavior Ontology is comparable with a meta-model. Using an ontology for this purpose has several advantages, such as easy integration with other ontologies (like domain ontologies) and the availability of advanced reasoning mechanisms.

4.1 Behavior Definitions

The first step of the VR-WISE behavior modeling process consists of building so-called *behavior definitions*. A behavior definition allows the designer to define the different behaviors for an object. Note that in our approach, the behaviors of an object are defined separately from the static structure of the object (its properties) and independent of how the behavior will be triggered. This will improve the reusability of behaviors within different VEs. In section 4.2, we will explain how to define the assignment and the triggering of behavior. Within the limited space of this paper it is not possible to give a very detailed explanation of the different modeling concepts provided. This is also not the purpose of this paper. A detailed presentation of modeling behavior in VR-WISE can be found in [9]. Here we concentrate on the ontology itself. The OWL Web Ontology Language [6] is used for expressing it. Figure 2 gives an overview of the relevant concepts ('bo' is the namespace used for the concepts and stands for 'behavior ontology'). In the following we will use italic to denote classes and relationships. In the Behavior Ontology, all modeling concepts for defining behavior are defined as subclasses of the class *BehaviorDefinitionElement*. A *BehaviorDefinition* contains a number of *BehaviourDefinitionElements* (*contains* relationship).

Actor. The main modeling concept is the *Actor*. An actor (e.g. a Door) represents an object involved in a behavior. The behaviors in which an actor is involved (e.g. OpenDoor, CloseDoor) is represented in the ontology by means of the *plays* relationship with *Behavior*. Because we separate the definition of

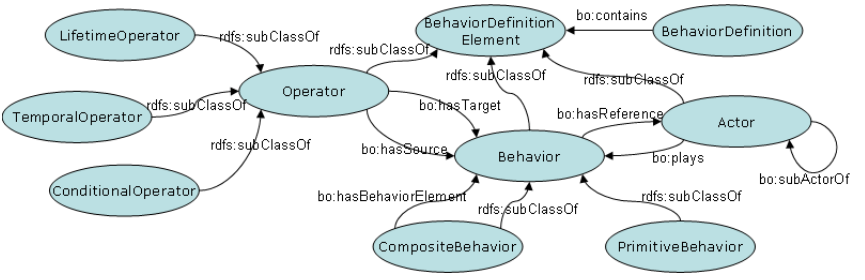


Fig. 2. Ontology Extraction

a behavior from the actual definition of (the structure of) an object, actors are used in the definition of a behavior instead of the actual object(s). An actor is a kind of abstract object. Furthermore, for an actor, we only specify the minimal properties required to have the specified behavior. This implies that each object that has those minimal properties can replace the actor and thus obtain the behavior defined for the actor. A generalization/specialization link can be defined between two actors. In that case, the child actor (e.g. a Sliding Door) inherits all the behavior defined for the parent actor (e.g. Door) and optionally adds additional behavior or overrides inherited behavior. In the ontology this is represented by means of the *subActorOf* relationship.

Behavior. We here focus on a particular type of behavior describing the movement of objects. We distinguish the following types of primitive behaviors: move, turn, roll, orient and position. They either change the position of an object or its orientation. These concepts (not shown in figure 2) are represented in the ontology as subclasses of *PrimitiveBehavior*, which is a subclass of *Behavior*.

The *move* can be used to express a change in the position of an object. To completely specify a move, a direction and a distance are needed. The direction can have one of the values: 'left', 'right', 'forward', 'backward', 'up' or 'down'. These simple directions can be combined to obtain more complex ones. The distance parameter (a value and a unit (e.g. meter)) expresses the distance to move. For a *turn*, the value for the direction can only be 'left' or 'right'. This is because a turn of an object is only possible around the top-to-bottom axis of the object. An angle parameter is needed to specify how much the object needs to be turned. A *roll* specifies a change in the orientation around the object's front-to-back axis in which case the value for direction can be 'left' or 'right', or around the left-to-right axis in which case the value for direction can be either 'forward' or 'backward'. By default, the directions specified for the primitive behaviors are the directions as perceived from the object's local reference frame. However, sometimes we want the object to do the movement 'as seen from' another object meaning that an external reference frame needs to be used. In the ontology, this is represented by means of the *hasReference* relationship.

Two alternative primitives have been defined for specifying movement. Using *position*, the position of an object can be directly changed by giving new coor-

dinates. Through the *passBy* relationship a number of intermediate positions can be given in order to obtain a smooth path following behavior. Alternatively an object can be given a position by means of spatial relations (e.g. in-front-of another object). The *orient* behavior allows giving the object a new orientation or alternatively the new orientation can be specified by means of an orientation relation (e.g. aligned-with another object). In the latter two behaviors, the reference object can be given by means of the *hasReference* relationship. Additional parameters (i.e. speed, ease-in, ease-out,...) can be specified by means of a value and a unit.

Composite behaviors can be defined by combining behaviors (either primitive or composite ones). This is represented in the ontology by the *hasBehaviorElement* relationship between *CompositeBehavior* and *Behavior*. The composite behaviors can be parameterized through the *hasParameter* relationship.

Operator. In order to achieve more complexity, behaviors can be combined in different ways with each other by means of operators (represented by the class *Operator*). For this purpose, we can use temporal operators (*TemporalOperator*), lifetime operators (*LifetimeOperator*) and conditional operators (*ConditionalOperator*). Every one of these operators has a source (s), defined by *hasSource*, and a target (t), defined by *hasTarget*, which have *Behavior* elements as a value.

Temporal operators allow synchronizing behaviors. They are based on the binary temporal relations. Some examples are *before(s, t, x)* (behavior s ends x seconds before the behavior t starts), *meets(s, t)* (behavior t starts immediately after the end of behavior s), *overlaps(s, t, x)* (behavior t starts x seconds before the end of behavior s), etc. Lifetime operators control the lifetime of a behavior which can be either enabled or disabled, and when it is enabled, it can be either active or passive. Some operators in this category are *enable(s, t)* (behavior t gets enabled when behavior s ends), *disable(s, t)* (behavior t is disabled just after behavior s ends), etc. A conditional operator allows controlling the flow of a behavior. By using a conditional operator, the behavior that will be invoked depends on the value of the conditional expression. Conditional expressions can be built using the standard mathematical and logical operators.

4.2 Behavior Invocations

After the behavior definitions are given, the next step consists of assigning the behaviors defined to the actual objects. In this step, the designer also specifies how the behaviors can be invoked (triggered). Using this approach, the interaction is separated from the actual definition of the behavior. It has the advantage that the same behavior can be associated to different objects and also that the same behavior can be triggered by means of different interactions depending on the situation. Figure 3 gives an overview of the most important classes and relationships in the ontology for describing the behavior invocations. Note that some of the classes from figure 2 are repeated to clearly represent the connection between both extractions of the ontology. They are represented with a dashed line. The modeling concepts for assigning behavior to objects and defining the triggering of the behaviors are defined with a class called *Behavior-*

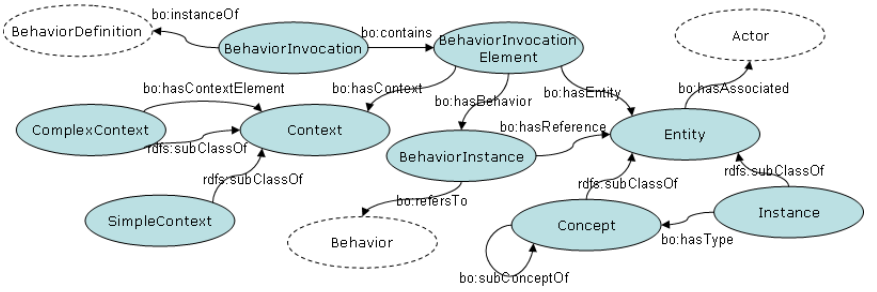


Fig. 3. Ontology Extraction

InvocationElement. This element connects an *Entity*, a *BehaviorInstance*, and a *Context*. These connections are defined by the relationships *hasEntity*, *hasBehavior* and *hasContext* respectively. A *BehaviorInvocation* consists of a number of *BehaviorInvocationElements* represented by the relationship *contains*. This element is an instance of a *Behavior* as can be seen by the *instanceOf* relationship.

Concept-Instance. The main modeling concepts are *Concept* and *Instance* both are subclasses of *Entity*. In the VR-WISE approach, the structure of the VE is expressed in terms of intuitive domain concepts (e.g. Gate), comparable to object types, and their instances. The concepts are brought into a hierarchical structure by means of the *subConceptOf* relationship. Every *Instance* has a *Concept* as type. This is represented in the ontology through the *hasType* relationship. An actor can be assigned to an entity through the *hasAssociated* relationship. With this, the entity obtains all the actors' behaviors. In the case of a concept, every instance of that concept will have all the behaviors defined for the actor (e.g. when Gate is associated with the actor Sliding Door, a Gate instance will has all behavior defined for the Sliding Door actor). In the case of an instance, only that particular instance will have all the behaviors of the actor (e.g. if FrontGate is a particular Gate instance and FrontGate is associated with the actor Sliding Door, only the FrontGate will have the behavior defined for Sliding Door). Entities can have multiple actors being assigned.

BehaviorInstance. A *BehaviorInstance* represents a reference to a particular *Behavior* (e.g. OpenDoor). This reference is formalized in the ontology through the *refersTo* relation. The entities acting as a reference-actor for a behavior can also be assigned using the *hasReference* relationship. Inside the *BehaviorInstance* element we can also specify the values for parameters of the behavior as well.

Context. The *Context* class in our ontology is used for describing the triggering mechanism. If the context evaluates to true, the behaviors connected to the context are triggered. The concept of context is taken from [4]. We distinguish between simple context and complex context. A simple context is represented by the *SimpleContext* class. An example of a simple context is: "the temperature in the environment is greater then 30". So, it is a statement that describes

some information about an entity. The entity is given by means of the *hasEntity* relation. The rest of a simple context is described by two more relations: a *hasRelationship* relation which refers to a context relation (described below), and a *hasValue* relation indicating the value of the relationship. In the example, "temperature" is the entity, "greater then" the relation and "30" the value.

For the context relations we provide a number of predefined relationships. These are the comparison relations, the localization relations, the interaction relations and the time relations. Note that other relationships can be added easily. The comparison relationships (e.g. *equals*, *greaterThen*, *smallerThen*,...) allow to test on properties of elements. The localization relationships like the *locatedIn* and the *locatedAt* relationship are provided to allow to test on the fact that a user (or any other object) is spatially located inside respectively nearby a particular object. Interaction relationships allow us to react to interactions either between the user and an object (like *onSelect*, *onClick*, *onTouch*,...) or between two objects (like *OnCollide*). Time relationships, based on the standard temporal relations, enable describing simulation time related context information.

Complex context statements, represented by the *ComplexContext* class in the ontology, can be built by combining a number of context statements (either simple or complex ones). This can be done using the *hasContextElement* relation. The (sub-) context statements are combined using the standard logical operators.

5 Implementation

A prototype tool, called OntoWorld, has been developed to support our approach. The tool enables a designer to make a conceptual design. We have extended the prototype tool with the Conceptual Specification Generator (CSG) [9]. This is a graphical diagram editor supporting the modeling of the behavior as described in this paper. The tool has been implemented as an extension to Microsoft Visio. It can be considered as a graphical interface for the specification phase (see section 3). Furthermore, OntoWorld allows specifying the desired mappings and finally generates a working VE, respectively step 2 and 3 of the design approach (see section 3). The code for the static scene is generated in an X3D [11] format together with ECMAScript fragments for the dynamical aspect. The semantic information by means of the ontologies can also be exported for use in different applications (e.g. search engines, reasoners,...).

6 Conclusion

In this paper we have discussed a behavior ontology which can be used to specify object behavior in a VE. Until now, modeling behavior was a very complex task. Creating a behavior specification based on our ontology may reduce the complexity of creating dynamic and interactive VEs. The ontology contains the concepts needed to describe the behavior independent of the objects itself, to assign the behaviors to the objects and indicate how they could be triggered. The ontology

is part of a set of upper ontologies that are used in the VR-WISE design approach to enable the modeling of VEs at a conceptual level. The ontologies enable cross-platform development of VEs and facilitate interoperability. Furthermore, the semantic information can be used for other purposes, e.g. searching in and semantic exploration of the VE.

Future work will focus on evaluating the modeling concepts by means of user experiments. Then, we extend the behavior ontology to support the design of more complex behaviors. For example, the modeling of forces will be considered, to obtain VR applications with a higher degree of physical correctness. We will also investigate how to model, at a high level, constraints on the object behavior.

Acknowledgements

This research is carried out in the context of the OntoBasis and the VR-DeMo project; both projects are funded by the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT). It is also partially funded by the FWO (Fund of Scientific Research - Flanders).

References

1. Bille, W., Pellens, B., Kleinermann, F. and De Troyer, O.: Intelligent Modelling of Virtual Worlds Using Domain Ontologies, In: Proceedings of the Workshop of Intelligent Computing (WIC), Mexico City, Mexico (2004) 272-279
2. Coyne, B. and Sproat, R.: WordsEye: An automatic text to scene conversion system, In: Proceedings of the International Conference on Computer Graphics and Interactive Technologies, Los Angeles, USA (2001) 487-496
3. De Troyer, O., Bille, W., Romero, R. and Stuer, P.: On Generating Virtual Worlds from Domain Ontologies, In: Proceedings of the 9th International Conference on Multimedia Modeling, Taipei, Taiwan (2003) 279-294
4. Dey, A.K., and Bowd G.D.: Towards a Better Understanding of Context and Context-Awareness. In: Proceedings of the Workshop on the What, Who, Where, When, and How of Context Awareness, The Hague, The Netherlands (2000) 1-6
5. Huang, Z., Eliens, A. and Viseer, C.: Implementation of a scripting language for VRML/X3D-based embodied agents, In: Proceedings of the Web3D 2003 Symposium, Saint Malo, France, S. Spencer (ed.) ACM Press (2003) 91-100
6. McGuinness, D.L., Harmelen, F.: OWL Web Ontology Language Overview, www.w3.org/TR/owl-features (2004)
7. Miller, J., Baramidze, G., Sheth, A., Fishwick, P.: Investigating Ontologies for Simulation Modeling. In: Proceedings of 37th Simulation Symposium (2004) 55-63
8. Otto, K.: Semantic Virtual Environments, In: the proceedings of the 14th international World Wide Web conference, ACM press, Chiba, Japan (2005) 1036-1037
9. Pellens, B., De Troyer, O., Bille, W. and Kleinermann, F.: Conceptual Modeling of Object Behavior in a Virtual Environment, In: Proceedings of Virtual Concept 2005, Publ. Springer-Verlag, Biarritz, France (2005) (accepted for publication)
10. Uschold, M. and Gruninger, M.: Ontologies: Principles, methods and applications. In: Knowledge Engineering Review, Vol. 11, No. 2 (1996) 93-155
11. Web 3D Consortium (Web3D): Extensible 3D (X3D) International Standard (2003) http://www.web3d.org/x3d/specifications/

Author Index

Abraham, Johnson 244
Adamus, Radoslaw 367
Ahlgren, Riikka 572
Albani, Antonia 582, 592
Albertoni, Riccardo 896
Albín, J.L. 327
Ali, Ali Shaikh 244
Allen, Gabrielle 294
Ambite, José Luis 30
An, Yuan 967
Aparicio, Ramiro 337
Athanasis, Nikolaos 1127
Avrithis, Yannis 977

Baloian, Nelson A. 1159
Balsters, H. 28, 109
Barrère, F. 537
Barrio, Rubén 836
Bédard, Yvan 999
Bell, David 856
Benjamins, Richard 846
Benzekri, A. 537
Bille, Wesley 1215
Billen, Roland 1066
Bixio, Franco 836
Blázquez, Mercedes 846
Botía Blaya, Juan A. 69
Brinkkemper, Sjaak 866
Broens, Tom 761
Buchanan, George 12
Busse, Susanne 1179

Cabaleiro, J.C. 327
Caballé, Santi 274
Campos, Cristina 552
Casanovas, Pompeu 846
Casellas, Núria 846
Casteleyn, Sven 700
Castells, Pablo 977
Catarci, Tiziana 562
Ceravolo, Paolo 809, 987
Chalmeta, Ricardo 552
Chang, Elizabeth 816, 916, 936, 957
Chang, Jae-Woo 1107
Chaudhri, Vinay K. 30
Cho, Young Ho 24
Choi, Joon Yeon 142
Choi, Ju-Ho 254
Choo, Hyunseung 347
Chung, Kwang-Sik 1097
Cîrstea, Corina 39
Clementini, Eliseo 1066
Coulson, Geoff 18
Craske, Gregory 5
Cullot, N. 1027
Cybula, Piotr 387

Damiani, Ernesto 809, 987
Daradoumis, Thanasis 274
da Silva, Alberto Rodrigues 516
de Cesare, Sergio 856
De Cock, Martine 1077
Dekate, Chirag 294
dela Cruz, Necito 636
Delgado, Jaime 836
De Maeyer, Philippe 1087
De Maria, Elisabetta 1
de Moor, Aldo 1190
Derballa, Volker 592
De Tré, Guy 1087
De Troyer, Olga 700, 1215
De Virgilio, Roberto 132
Dietz, Jan L.G. 688
Dillon, Tharam S. 816, 916, 936, 957
Dimov, Stefan S. 1149
Doi, Norihisa 59
Donnay, Jean-Paul 1117
Dou, Dejing 35
Durand, David 16
Duval, Erik 1169

Encheva, Sylvia 79
Evans, Ken 646

Fankhauser, Peter 926
Feng, Ling 936
Fernández, Miriam 977
Ferri, Fernando 1009
Fikes, Richard 30

Fisher, Peter F. 1056
Formica, Anna 1009
Fortier, Andrés 176
Frampton, Keith 431
Freiheit, Jörn 26
Fujii, Kunihiro 876
Fukazawa, Yusuke 876
Funk, Caroline 215

Gal, Avigdor 947
García, Clemente, Félix J. 69
Gašević, Dragan 1169
Georget, Sebastien 314
Gómez-Pérez, Asunción 906
Gómez Skarmeta, Antonio F. 69
Goncalves, Marlene 790
Gordillo, Silvia 176
Gouvas, Panagiotis 452
Greenhalgh, Chris 284
Grifoni, Patrizia 1009
Grün, Ch. 206
Gulla, Jon Atle 473

Haase, Peter 906
Habela, Piotr 377
Hafner, Michael 506
Hallot, Frédéric 771
Halpin, Terry 676
Ham, Eunmi 1200
Hampshire, Alastair 284
Han, Dongsoo 461
Hartmann, Jens 906
Heimrich, Thomas 7
Henderson, Peter 39
Henricksen, Karen 122, 626
Herrero, Pilar 337, 397
Higel, Steffen 49
Hinze, Annika 12, 152
Hofreiter, Birgit 408
Honiden, Shinichi 14
Hoppenbrouwers, S.J.B.A. 666, 720
Huang, Dayong 294
Huedo, Eduardo 234
Huemer, Christian 408
Huh, Eui-Nam 347
Huitema, G.B. 28, 109
Hussain, Farookh Khadeer 957
Hutanu, Andrei 294
Hwang, Chong-Sun 1097

Indulska, Jadwiga 122, 626
Ingvaldsen, Jon Espen 473

Jabeur, Nafaâ 99
Jaekel, Frank Walter 552
Jarrar, Mustafa 613
Jaśkowski, Tomasz 886
Jenkins, Jessica 30
Jeong, Chang-Sung 254
Jonker, Wim 526
Jovanović, Jelena 1169
Jung, Doris 152
Jung, Hyosook 211
Jung, SoonYoung 1097

Kaczmarski, Krzysztof 377
Kalabokidis, Kostas 1127
Kamel, M. 537
Kang, Myong, 89
Kavouras, Marinos 1137
Kayed, Ahmad 826
Keet, C. Maria 603
Kerre, Etienne E. 1077
Khodawandi, Darius 741
Kim, Chang Ouk 24
Kim, Jong-Sun 22
Kim, Kwang-Hoon 485
Kim, Yoo-Jung 347
Kirchner, Holger 156
Kirsche, Thomas 9
Kiryakov, Atanas 846
Kłeczek, Dariusz 886
Kleinermann, Frederic 1215
Knežević, Predrag 526
Kong, Ki-Sik 1097
Kopecký, Jacek 229
Koschmider, Agnes 495
Kozankiewicz, Hanna 377, 387
Krummenacher, Reto 229
Kuhmünch, Christoph 215
Kuhn, Werner 1020
Kuijpers, Bart 1087
Kuliberda, Kamil 367
Kurakake, Shoji 876
Kwak, Choon Jong 24
Kwak, Myungjae 461
Kwon, Yong-Won 254

Laborde, R. 537
Lagos, Nikolaos 1149

Laplanche, François 1117
Larrivée, Suzie 999
Lavirotte, Stéphane 225
Lawrence, Dave R. 9
Lee, Hong Joo 142
Lee, Sung-Young 33
Lee, Wang-Chien 1107
Lee, Young-Koo 33
Lehti, Patrick 926
LePendu, Paea 35
Leprevost, Franck 314
Lewis, David 49
Lingrand, Diane 225
Llorente, Ignacio M. 234
Llorente, Silvia 836
Logé, Christophe 16
Longo, Isabella 836
Luo, Jim 89
Lycett, Mark 856

MacLaren, Jon 294
Małanij, Rafał 886
Markkula, Jouni 572
Martínez, D.R. 327
Martínez, Pérez, Gregorio 69
Martins, Paula Ventura 516
Matsuzaki, Kazutaka 14
McFadden, Ted 626
Meersman, Robert 800
Mendling, Jan 506
Mentzas, Gregoris 452
Mertins, Kai 552
Mevius, Marco 495
Milano, Diego 562
Minout, Mohammed 1037
Mishra, Sunil 30
Montagnat, Johan 314
Montanari, Angelo 1
Montero, Rubén S. 234
Montrose, Bruce 89
Motelet, Olivier 1159
Moulin, Bernard 99
Münch, Susanne 26
Muslea, Maria 30
Müssigmann, Nikolaus 582
Mylonas, Phivos 977
Mylopoulos, John 967

Naganuma, Takefumi 876
Nagypál, Gábor 780
Nasser, B. 537
Ngoc, Kim Anh Pham 33
Niedermeier, Christoph 215
Noh, Hack-Youp 22

O'Sullivan, Timothy 186

Paik, Woojin 1200
Painho, Marco 1020
Palma, Raúl 906
Paniagua, Claudi 274
Papaleo, Laura 896
Park, Byungchul 347
Park, KwangJin 1097
Park, Seongbin 211
Park, Sung Joo 142
Parmee, Ian 244
Pasquasy, Fabien 1117
Pellens, Bram 1215
Pena, T.F. 327
Penttilä, Jari 572
Pepels, Betsy 656
Perepletchikov, Mikhail 431, 442
Pérez, María S. 337, 397
Perry, Nicolas 552
Pierson, Eric John 636
Pires, Paulo 1020
Pitikakis, Marios 896
Plasmeijer, Rinus 656
Plessers, Peter 700
Poblet, Marta 846
Pokraev, Stanislav 526
Porter, Barry 18
Pouliot, Jacynthe 999
Prastacos, Poulicos 1137
Pröll, B. 206
Proper, H.A. (Erik) 666, 720, 730
Puder, Arno 20

Radenkovic, Milena 264
Rafanelli, Maurizio 1009
Rajugan, R. 936
Rana, Omer F. 244
Reichelt, Katrin 7
Reinberger, Marie-Laure 1205
Rerrer, Ulf 196
Retschitzegger, W. 206
Rivera, F.F. 327
Robbiano, Francesco 896
Roch, Jean-Louis 314

Rodríguez, Eva 836
Romero, Raul 1215
Rønneberg, Harald 473
Rossi, Gustavo 176
Rowińska, Edyta 886
Rüetschi, Urs-Jakob 1047
Rusch, Hendrik 7
Ryan, Caspar 5, 431, 442
Ryu, So-Hyun 254

Sainte, Jean-Christophe 1117
Salvadores, Manuel 337, 397
Sánchez, Alberto 337, 397
Sattler, Kai-Uwe 7
Scannapieco, Monica 562
Schanzenberger, Anja 9
Schmidt, Rainer 421
Schmidt, Stefan 816
Schockaert, Steven 1077
Scholl, Michel 166
Schöttle, Hendrik 26
Schröder, Thomas 7
Schwinger, W. 206
Seo, Yoon Ho 24
Setchi, Rossitza M. 1149
Shackelford, Mark 244
Shim, Jaeyong 461
Sidhu, Amandeep S. 916
Sijanski, Grozdana 26
Song, MoonBae 1097
Sotnykova, A. 1027
Soulakellis, Nikolaos 1127
Spagnuolo, Michela 896
Spanaki, Maria 1137
Spyns, Peter 710, 1205
Steele, Robert 816
Stencel, Krzysztof 387
Strang, Thomas 229
Studdert, Richard 186
Studer, Rudi 906
Su, Xiaomeng 473
Suárez-Figueroa, M. Carmen 906
Subieta, Kazimierz 367, 377, 387
Sure, York 906
Szirbik, N.B. 109

Takada, Shingo 59
Tang, Yan 800
Tari, Zahir 442
Thilliez, Marie 166
Tigli, Jean-Yves 225
Timpf, Sabine 1047
Tomai, Eleni 1137
Torlone, Riccardo 132
Tumin, Sharil 79
Turchenko, Volodymyr 357

Um, Jung-Ho 1107
Uribe, Tomas 30
Usanvasin, Sasiporn 59

Vaitis, Michail 1127
Vallet, David 977
Van de Weghe, Nico 1087
van der Weide, Th.P. 666, 720, 730
Vangenot, C. 1027
Varrette, Sebastien 314
Vasilakis, George 896
Verbert, Katrien 1169
Verginadis, Giannis 452
Vetter, Claus 3
Vidal, María-Esther 790
Viviani, Marco 809
Voisard, Agnès 166
Voudouris, Vlasios 1056

Wac, Katarzyna 751
Wade, Vincent 49
Wasilewski, Marcin 886
Werner, Thomas 3
Wieringa, Roel 526
Wietrzyk, Bartosz 264
Wislicki, Jacek 367
Wombacher, Andreas 526
Wood, Jo 1056

Xhafa, Fatos 274
Xu, Lai 866

Yang, Guizhen 30
Yang, Jingtao 39
Yoneki, Eiko 304
Yoo, Ji-Yoon 22
Yoon, Jung Uk 24
Yoshioka, Nobukazu 14

Zangl, Fabrice 26
Zantoni, Marco 1
Zhang, Chongjie 294
Zimányi, Esteban 1037