

Lecture Notes in Artificial Intelligence 4183

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Jérôme Euzenat John Domingue (Eds.)

Artificial Intelligence: Methodology, Systems, and Applications

12th International Conference, AIMS 2006
Varna, Bulgaria, September 12-15, 2006
Proceedings

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Jérôme Euzenat
INRIA Rhône-Alpes
655 avenue de l'Europe, 38330 Montbonnot Saint-Martin, France
E-mail: jerome.euzenat@inrialpes.fr

John Domingue
The Open University, Knowledge Media Institute
Milton Keynes MK7 6AA, UK
E-mail: j.b.domingue@open.ac.uk

Library of Congress Control Number: 2006932035

CR Subject Classification (1998): I.2, H.4, F.1, H.3, I.5

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN	0302-9743
ISBN-10	3-540-40930-0 Springer Berlin Heidelberg New York
ISBN-13	978-3-540-40930-4 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
springer.com

© Springer-Verlag Berlin Heidelberg 2006
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11861461 06/3142 5 4 3 2 1 0

Preface

The 12th Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA 2006) was held in Varna on the Black Sea coast during September, 12–15, 2006. The AIMSA conference series has provided a biennial forum for the presentation of artificial intelligence research and development since 1984. The conference, which is held in Bulgaria, covers the full range of topics in artificial intelligence and related disciplines and provides an ideal forum for international scientific exchange between Central/Eastern Europe and the rest of the world. The 2006 edition perpetuates this tradition.

For AIMSA 2006, we wanted to place special emphasis on a specific phenomenon that affects all areas of artificial intelligence: the application and leverage of artificial intelligence technology in the context of human collaboration which today is mediated by the Web. Artificial intelligence is used to support human communication in a wide variety of ways. For example, reasoning over the Semantic Web, analyzing relationships between people, enhancing the user experience by learning from their behavior, applying natural language to large multilingual corpora, planning a combination of Web services, and adapting and personalizing educational material. A plethora of artificial intelligence techniques are amenable to facilitating communication on the Web. Moreover, these techniques are not deployed in isolation but are typically combined with results from other disciplines such as the social sciences, discrete mathematics, network computing, or cryptography.

However, as its name indicates the conference is also dedicated to artificial intelligence in its entirety. As such, AIMSA remains a generalist artificial intelligence conference with tracks on constraint satisfaction, agents, ontology, decision support, natural language processing and machine learning. The Web effect has not created its own new sub-discipline of artificial intelligence but rather affects all of its sub-areas.

Eighty-one interesting papers were submitted to the conference coming from 27 different countries representing 5 continents. Each paper was reviewed by more than three independent reviewers on average. The Program Committee selected for the conference program 29 contributions out of which 28 are included in the current volume. We would like to thank the AIMSA Program Committee and the additional reviewers for their hard work in assessing paper quality.

In addition to the selected papers, the AIMSA conference featured two workshops on “Natural Language Processing for Metadata Extraction” and “Semantic Web and Knowledge Technologies Applications.” AIMSA 2006 had two invited speakers who, signs of the time, addressed intelligent technologies in the large. Enrico Motta from the Open University considered the challenges raised by dealing with increasing amount of semantic markup on the Web and Fabio Ciravegna

from the University of Sheffield discussed those of acquiring and sharing knowledge in large organizations with reports from large-scale case studies.

July 2006

Jérôme Euzenat
John Domingue

Organization

Conference Chair	John Domingue (Open University, UK)
Program Chair	Jérôme Euzenat (INRIA Rhône-Alpes, France)
Organizing Chair	Danail Dochev (IIT-BAS, Bulgaria)
Organization Committee	Gennady Agre Ivo Marinchev Kamenka Staykova Violeta Magerska

Program Committee

Gennady Agre (Bulgarian Academy of Sciences, Sofia)
Lina Al-Jadir (EPFL, Lausanne)
Leila Amgoud (IRIT, Toulouse)
Anupriya Ankolekar (AIFB-University of Karlsruhe)
Grigoris Antoniou (ICS-FORTH, Heraklion)
Nathalie Aussenac-Gilles (IRIT, Toulouse)
Jean-François Baget (INRIA Rhône-Alpes, Montbonnot)
Sean Bechhofer (University of Manchester)
Richard Benjamins (iSOCO, Barcelona)
Bettina Berendt (Humboldt University Berlin)
Petr Berka (University of Economics, Prague)
Abraham Bernstein (University of Zurich)
Kalina Bontcheva (University of Sheffield)
Omar Boucelma (LSIS-Université Aix-Marseille 3)
Paulo Bouquet (University of Trento)
Joost Breuker (University of Amsterdam)
Liliana Cabral (The Open University, Milton Keynes)
Diego Calvanese (Free University of Bozen-Bolzano)
Tiziana Catarci (University of Rome 1)
Jean Charlet (Assistance Publique-Hopitaux de Paris)
Frithjof Dau (TU Dresden)
Jos De Bruyn (DERI-University of Innsbruck)
Yves Demazeau (Leibniz-IMAG, Grenoble)
Christo Dichev (Winston-Salem State University)
Ying Ding (DERI-University of Innsbruck)
Pavlin Dobrev (ProSyst Labs, Sofia)
Danail Dochev (Bulgarian Academy of Sciences, Sofia)

Martin Dzubor (The Open University, Milton Keynes)
Peter Eklund (University of Wollongong, Australia)
Dieter Fensel (DERI-University of Innsbruck, National University Ireland
Galway)
Frederico Freitas (Universidade Federal de Pernambuco)
Aldo Gangemi (ISTC-CNR, Rome)
Jennifer Golbeck (University of Maryland)
Christine Goldbreich (Université de Rennes 1)
Asunción Gómez-Pérez (Universidad Politécnica de Madrid)
Marko Grobelnik (Jozef Stefan Institute, Ljubljana)
Siegfried Handschuh (DERI-National University of Ireland, Galway)
Andreas Herzig (IRIT, Toulouse)
Pascal Hitzler (AIFB-University of Karlsruhe)
Philippe Jorrand (Leibniz-IMAG, Grenoble)
Vipul Kashyap (Partners HealthCare System)
Irena Koprinska (University of Sydney)
Robert Kremer (University of Calgary)
Atanas Kyriakov (Sirma - Ontotext Lab, Sofia)
Alain Léger (France Telecom R&D)
Raphael Malyankar (Arizona State University)
Massimo Marchiori (W3C and University of Venice)
Pierre Marquis (CRIL-Université d'Artois)
John-Jules Meyer (Utrecht University)
Michele Missikoff (IASI-CNR, Rome)
Riichiro Mizoguchi (Osaka University)
Boris Motik (University of Manchester)
Enrico Motta (The Open University, Milton Keynes)
Marie-Laure Mugnier (LIRMM, Montpellier)
Amedeo Napoli (LORIA, Nancy)
Wolfgang Nejdl (L3S-University of Hannover)
Borys Omelayenko (Vrije Universiteit Amsterdam)
Massimo Paolucci (DoCoMo European Laboratories, Munich)
Radoslav Pavlov (Bulgarian Academy of Sciences)
Christoph Quix (RWTH Aachen)
Marie-Christine Rousset (LSR-Université Joseph-Fourier, Grenoble)
Michèle Sebag (LRI, Orsay)
Luciano Serafini (ITC, Trento)
Pavel Shvaiko (University of Trento)
Carles Sierra (IIIA, Barcelona)
Michael Sintek (DFKI, Kaiserslautern)
Helena Sofia Pinto (IST-Technical University of Lisbon)
Giorgos Stamou (NTUA, Athens)
Umberto Straccia (CNR, Pisa)
Heiner Stuckenschmidt (University of Mannheim)
Gerd Stumme (Universität Kassel)

York Sure (University of Karlsruhe)
Valentina Tamma (University of Liverpool)
Sergio Tessaris (Free University of Bozen-Bolzano)
Raphaël Troncy (CWI, Amsterdam)
Petko Valtchev (University of Montréal)
Laure Vieu (IRIT, Toulouse)

Additional Reviewers

Shadi Abras	Valeri Ilchev	Guillaume Piolle
Sudhir Agarwal	Jason Jung	Jean-François Puget
Eva Armengol	Arno Kamphuis	Katharina Reinecke
Philippe Besnard	Christoph Kiefer	Olivier Roussel
Sebastian Blohm	Sébastien Laborie	Giuseppe Santucci
Luka Bradeako	Steffen Lamparter	Omair Shafiq
Janez Brank	Christophe Lecoutre	Joseph Sindhu
Augusto Costa	Freddy Lecue	Lucia Specia
Jean-Yves Delort	Jiwen Li	Heiko Stoermer
Blaz Fortuna	Yaoyong Li	Vassil Vassilev
Dorian Gaertner	Knud Möller	Holger Wache
Daniel Giacomuzzi	Michele Pasin	Antoine Zimmermann
Antoon Goderis	Damien Pellier	
Tudor Groza	Jérôme Pierson	

Sponsoring Institutions

Bulgarian Artificial Intelligence Association
Institute of Information Technologies (IIT-BAS)

Table of Contents

Invited Talks

Exploiting Large-Scale Semantics on the Web.....	1
<i>Enrico Motta</i>	
Acquiring and Sharing Knowledge in Large Organizations: Issues, Requirements and Methodologies	2
<i>Fabio Ciravegna</i>	

Agents

Property Based Coordination.....	3
<i>Mahdi Zargayouna, Julien Saunier Trassy, Flavien Balbo</i>	
A Formal General Setting for Dialogue Protocols.....	13
<i>Leila Amgoud, Sihem Belabbès, Henri Prade</i>	
OCC's Emotions: A Formalization in a BDI Logic.....	24
<i>Carole Adam, Benoit Gaudou, Andreas Herzig, Dominique Longin</i>	

Constraints and Optimization

A Boolean Encoding Including SAT and n-ary CSPs.....	33
<i>Lionel Paris, Belaïd Benhamou, Pierre Siegel</i>	
A Constructive Hybrid Algorithm for Crew Pairing Optimization	45
<i>Broderick Crawford, Carlos Castro, Eric Monfroy</i>	
Using Local Search for Guiding Enumeration in Constraint Solving.....	56
<i>Eric Monfroy, Carlos Castro, Broderick Crawford</i>	

User Concerns

Study on Integrating Semantic Applications with Magpie	66
<i>Martin Dzbor, Enrico Motta</i>	
N-Gram Feature Selection for Authorship Identification	77
<i>John Houvardas, Efsthios Stamatatos</i>	

Incorporating Privacy Concerns in Data Mining on Distributed Data	87
<i>Huizhang Shen, Jidi Zhao, Ruipu Yao</i>	

Decision Support

Multiagent Approach for the Representation of Information in a Decision Support System	98
<i>Fahem Kebair, Frédéric Serin</i>	
Flexible Decision Making in Web Services Negotiation	108
<i>Yonglei Yao, Fangchun Yang, Sen Su</i>	
On a Unified Framework for Sampling With and Without Replacement in Decision Tree Ensembles	118
<i>José María Martínez-Otzeta, Basilio Sierra, Elena Lazkano, Ekaitz Jauregi</i>	

Models and Ontologies

Spatio-temporal Proximities for Multimedia Document Adaptation	128
<i>Sébastien Laborie</i>	
Deep into Color Names: Matching Color Descriptions by Their Fuzzy Semantics	138
<i>Haiping Zhu, Huajie Zhang, Yong Yu</i>	
Developing an Argumentation Ontology for Mailing Lists	150
<i>Colin Fraser, Harry Halpin, Kavita E. Thomas</i>	

Machine Learning

Clustering Approach Using Belief Function Theory	162
<i>Sarra Ben Hariz, Zied Elouedi, Khaled Mellouli</i>	
Machine Learning for Spoken Dialogue Management: An Experiment with Speech-Based Database Querying	172
<i>Olivier Pietquin</i>	
Exploring an Unknown Environment with an Intelligent Virtual Agent . . .	181
<i>In-Cheol Kim</i>	

Ontology Manipulation

Case-Based Reasoning Within Semantic Web Technologies	190
<i>Mathieu d'Aquin, Jean Lieber, Amedeo Napoli</i>	

A Proposal for Annotation, Semantic Similarity and Classification of Textual Documents	201
<i>Emmanuel Nauer, Amedeo Naponi</i>	
Cooking an Ontology	213
<i>Ricardo Ribeiro, Fernando Batista, Joana Paulo Pardal, Nuno J. Mamede, H. Sofia Pinto</i>	

Natural Language Processing

Methodology for Bootstrapping Relation Extraction for the Semantic Web	222
<i>Maria Tchalakova, Borislav Popov, Milena Yankova</i>	
Using Verbs to Characterize Noun-Noun Relations	233
<i>Preslav Nakov, Marti Hearst</i>	

Applications

BEATCA: Map-Based Intelligent Navigation in WWW	245
<i>Mieczysław A. Kłopotek, Krzysztof Ciesielski, Dariusz Czerski, Michał Dramiński, Sławomir T. Wierchoń</i>	
Model-Based Monitoring and Diagnosis Chip for Embedded Systems	255
<i>Satoshi Hiratsuka, Hsin-Hung Lu, Akira Fusaoka</i>	
A Knowledge-Based Approach for Automatic Generation of Summaries of Behavior	265
<i>Martin Molina, Victor Flores</i>	
INFRAWEBs Designer – A Graphical Tool for Designing Semantic Web Services	275
<i>Gennady Agre</i>	
Author Index	291

Exploiting Large-Scale Semantics on the Web

Enrico Motta

Open University, Milton Keynes, UK

There is a lot of evidence indicating that the semantic web is growing very rapidly. For example, an IBM report published last year indicated a 300% increase between 2003 and 2004 and pointed out that the rate of growth of the semantic web is mirroring that of the web in its early days. Indeed, repositories such as Swoogle already contain thousands of ontologies and hundreds of millions of RDF triples. Thus, a large scale semantic web is rapidly becoming a reality and therefore we are quickly reaching the point where we can start thinking about a new generation of intelligent applications, capable of exploiting such large scale semantic markup. Of course, while the semantic web provides an exciting opportunity, it also introduces very complex challenges. For example, the available semantic markup is extremely heterogeneous both with respect to its ontological profile and also to its degree of quality and to the level of trust that can be assigned to it. These features of the semantic web introduce an element of complexity, which was absent from traditional knowledge-based systems, where data quality was under the control of the developers, and provenance and heterogeneity did not apply. In my talk I will discuss these issues in some detail, and in particular I will describe the emerging semantic landscape and highlight some of the distinctive features characterizing the new generation of applications, which will be enabled by a large scale semantic web. In my presentation I will also present some concrete initial prototypes of this new generation of semantic applications, which exploit available large scale web semantics, to provide new ways to support question answering, information extraction and web browsing.

Acquiring and Sharing Knowledge in Large Organizations: Issues, Requirements and Methodologies

Fabio Ciravegna

University of Sheffield, UK

Efficient and effective Knowledge Acquisition and Sharing are of vital importance for large organizations. Complex human and technical aspects make them a complex issue. In this talk I will describe and discuss requirements, tools and methodologies for Knowledge Acquisition and Sharing in large organizations. I will use examples from the aerospace and automotive domain to describe and discuss real world requirements, including those related to human factors and technical issues. Then I will describe techniques from the field of Human Language Technologies and the Semantic Web that can be used for addressing those requirements. I will then present examples of real world applications developed in the aerospace domain. Finally, I will discuss some future trends.

Property Based Coordination

Mahdi Zargayouna^{1,2}, Julien Saunier Trassy², and Flavien Balbo^{1,2}

¹ Inrets - Gretia, National Institute of Transportation Research and their Security,
2, av. du Général Malleret-Joinville,
F-94114 Arcueil cedex

² Lamsade, Paris Dauphine University.
Place Maréchal de Lattre de Tassigny,
75775 Paris Cedex 16, France

{zargayou, balbo, saunier}@lamsade.dauphine.fr

Abstract. For a multiagent system (MAS), coordination is the assumption that agents are able to adapt their behavior according to those of the other agents. The principle of Property Based Coordination (PBC) is to represent each entity composing the MAS by its observable properties, and to organize their perception by the agents. The main result is to enable the agents to have contextual behaviors. In this paper, we instantiate the PBC principle by a model, called EASI -Environment as Active Support of Interaction-, which is inspired from the Symbolic Data Analysis theory. It enables to build up an interaction as a connection point between the needs of the initiator, those of the receptor(s) and a given context. We demonstrate that thanks to PBC, EASI is expressive enough to instantiate other solutions to the connection problem. Our proposition has been used in the traveler information domain to develop an Agent Information Server dynamically parameterized by its users.

1 Introduction

One of the basic problems for the designer of a multiagent system (MAS) is the connection problem [4]. In [10], the authors define the connection problem as “finding the other agents that have the information or the capabilities that you need”. In order to solve this problem for the Internet, they use middle-agents. The authors define a middle-agent as an entity that is neither a requester nor a provider of a service but which participates in the interaction between the requester and the provider: a requester has preferences and a provider has capabilities. This interaction model, with in the one hand the capability of a provider and on the other the preferences of a requester, is called “Capability-Based-Coordination” (CBC). Other solutions are proposed to solve the connection problem when some organizational rules are present and when the connection problem depends on these rules. The AGR (Agent, Group, Role) model [5] proposes agents that are defined as an active communicating entity that plays roles (abstract representation of the agent’s functionalities) within a group. However, when the connection problem embeds other *ambient* criteria, which we call a context, a new interaction relation has to be defined. Ambient criteria concern conditions that correspond neither to the initiator nor to the receptor of the interaction. Indeed, when the interaction is not a binary but a multi-partite relation, being guarded by contextual conditions, a special model is to be built, considering a collective status of the MAS.

The aggregation of the individual status of the agents inside the MAS into a collective one is one of the motivations for the emergence of the multiagent environment as an explicit entity. Several directions are followed in modeling multiagent environments. Some researches focus essentially on the dynamics of the environment [6,11], and others focus on the modeling of new interaction patterns thanks to the presence of the environment as an explicit entity. For instance, the stigmergy allows agents to leave traces in the environment, which can be used by others to guide their actions. New kinds of interaction recently appeared, like the overhearing and the mutual awareness [7,8,9,12]. They envision new types of interaction that enable agents, that are not necessarily protagonists of the interaction (neither emitter nor receptor of a message), to participate in the interaction. These are propositions that investigate alternatives to the traditional messages' broadcast to every agent present in the system.

Our approach generalizes the modeling of interactions inside the environment. We propose a description of the environment based on the principle of observability, instantiating this way the principle of Property Based Coordination (PBC). We define the PBC as a coordination principle for multiagent systems in which: (i) Every entity composing the system, including the agents, exhibits observable properties, (ii) Agents use the observable properties to manage the interactions, perceptions and actions inside the system. Each entity in the environment is then described uniformly by a set of observable properties. Agents being in the center of the MAS modeling, they manage the interactions by specifying the conditions (the context) of the interaction. The CBC and the AGR model are sub-cases of the PBC: the agents exhibit respectively their capabilities and preferences (CBC) and their groups and role (AGR) as observable properties; and of course the dyadic interaction is also a sub-case (identifiers are the observable properties in this case). In order to describe a detailed proposition, we propose a model, that we called EASI (Environment as an Active Support of Interaction). It supports the PBC principle and its formulation is widely inspired from the Symbolic Data Analysis (SDA) paradigm [1]. This model proposes to share the interactions inside the environment and enables the expression of different patterns of interaction. We believe it is the most straightforward way to instantiate the PBC principle.

A model that integrates the other interaction patterns is important because it provides a general framework, that has to be expressive enough, to enable designers to express different configurations, different application scenarios, within the same model. *Tuplespaces* systems, with Linda [3,2] as a first implementation, resembles to our proposition. They provide a shared collection of data (tuples), and a set of operations (read, write, take) on the collection to manipulate the data. However, the agents need to know beforehand, the location of a tuplespace to interact in. In addition, the expressivity of the querying process does not permit to match different facts e.g. to condition the perception of a tuple to the presence of an other tuple, whereas in EASI, agents are able to express such a condition.

The remainder of this paper is organized as follows: section 2 presents the EASI model. Section 3 details interactions features that our model enables to express. Section 4 shows the use of our model for a traveler Information System. We finally draw general conclusions and perspectives in section 5.

2 Environment as Active Support of Interaction

2.1 Introduction

EASI is a model that instantiates the PBC principle, it proposes an environment model that enables to share the interactions. The problem, when all the interactions are in common, is to enable the agents to find those they are interested in. The formulation of the model is inspired from the Symbolic Data Analysis (SDA) theory [1]. SDA is aimed at discovering *hidden* knowledge within large data sets, modeling both qualitative and quantitative data, which are clustered via the so-called symbolic objects. A symbolic object is a symbolic representation of a subset of entities materialized by a set of conditions that these entities have to satisfy. In our work, the entities composing the environment are agents (active entities) and objects. Both are described via observable descriptions. The reification of the active behavior of agents is their exclusive possibility to *select* their interlocutors by defining autonomously the symbolic objects (*filters* in the model). Depending on the considered MAS, other classifications for the entities are possible; for instance messages, traces, events, services etc. But our classification is the most general that remains consistent with the multiagent paradigm. Indeed, the only finalistic entities (i.e. pursuing an objective) in the MAS are the agents, whereas every entity that is deterministic is considered as an object. Besides, the possibility to manage the properties' privacy i.e. to hide or to exhibit them, enables to model messages (e.g. exhibiting the header and hiding the body of the message), basic services (exhibiting the 'invokable' operations of the service) etc.

2.2 EASI: Basic Definitions

Definition 1 (Environment). $E = \langle \Omega, D, P, F \rangle$ where:

- Ω is the set of entities, it contains the set of agents (A) and the set of objects (O).
 $\Omega = A \cup O$.
- $P = \{P_i | i \in I\}$, P is the set of observable properties.
 $P_i : \Omega \rightarrow d_i \cup \{\text{null}, \text{unknown}\}$. P_i is an application which, for an entity, gives the value for the corresponding property.
- $D = \prod_{i \in I} d_i$ with d_i the description domain of P_i .
- F is the set of filters. A filter is a set of conditions defining the interaction context inside the environment.

The basic component of our model is an entity. Each entity is described by symbolic variables (observable properties); I contains the indices ranging over P . If an entity does not have a value for a given P_i (this property does not exist for this entity, e.g. the property *radius* for a triangle), then its value is *null*; if the property exists but does not have a value (hidden by the corresponding entity), its value is *unknown*. Since agents are autonomous, each agent is responsible for the update of its own properties, the latter could be an information about the agent's position in the environment, an activity indicator etc. Each d_i (description domain of P_i) can be quantitative, qualitative as well as a finite data set. A property P_i of an agent a can thus have a value d_i , a can hide it

($P_i(a) = \text{unknown}$) or it can be undefined for the corresponding agent ($P_i(a) = \text{null}$). The case where an agent hides a given property is temporary and could dynamically be changed during the execution, but the information about the non-definition of P_i is structural and clearly clusters the set A of agents in several categories.

Definition 2 (Agent Category). $A_C \in A$ is an Agent Category $\iff \forall a_i \in A_C, \forall P_j \in P$, if $P_j(a_i) \neq \text{null} \implies \nexists a_k \in A_C | P_j(a_k) = \text{null}$

Reduced to this definition, agent categories define a Galois-lattice (cf. Fig.1), with g and h are the Galois correspondences.

$$g : \mathcal{P}(A) \rightarrow \mathcal{P}(P), g(A_C) \mapsto \{P_j \in P | \forall a_i \in A_C, P_j(a_i) \neq \text{null}\}$$

$$h : \mathcal{P}(P) \rightarrow \mathcal{P}(A), h(P_C) \mapsto \{a_i \in A | \forall P_j \in P_C, P_j(a_i) \neq \text{null}\}$$

The top of the lattice represents the agents (may be virtual) which exhibit only null values and the bottom represents the (virtual) agents which exhibit no null values. The designer could define a new top by defining a minimal property (typically an identifier).

Definition 3 (Minimal property of an agent). $P_j \in P$ is a minimal property $\iff \forall a_i \in A, P_j(a_i) \neq \text{null}$

The same categorization can be applied to the objects O , and provides then a typology of objects present in the system. This information (together with D , the description domains) can be exhibited as a meta-information in the environment, which could be used by the newly coming agents in order to know the interaction possibilities inside the MAS.

In SDA, a symbolic object puts together (in the same ‘class’) entities according to their description. In order to model an interaction, more than one kind of entities (agents and objects) have to be gathered. A filter f is then defined as a conjunction of symbolic objects (which is a symbolic object too). A filter connects the description of an agent to

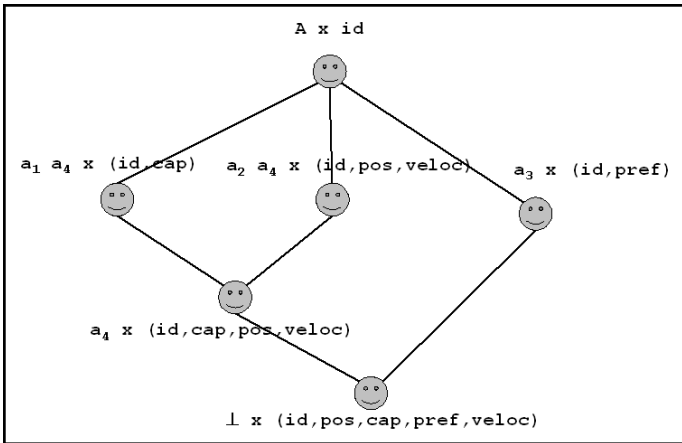


Fig. 1. Agent Categories Galois-lattice

the *context* that it perceives; a context is a state of the environment at a given moment e.g. the presence of a certain agent in the vicinity of a given node. Thus, the definition of a filter is:

Definition 4 (Filter). $f : A \times O \times \mathcal{P}(\Omega) \rightarrow \{true, false\}$

$$f(a, o, C) = \bigwedge_{i \in I_a} [P_i(a)R_iV_i] \bigwedge_{i \in I_o} [P_i(o)R_iV_i] \bigwedge_{i \in I_C} (\bigwedge_{c \in C} [P_i(c)R_iV_i])$$

$$f(a, o, C) \implies perceive(a, o)$$

$I_a \subset I$ (respectively I_o and I_C) contains the indices ranging over P that are used in f as selection criteria for the agent (respectively the object and the context). R and V are respectively the binary operators and the values of the descriptions that define the conditions to be held by a , o and C . A filter is the intention definition of the relation between a particular entity a and other entities c in C according to its own description and those of each c , it conditions the perception of a certain o by a . For instance, consider the following filter: $f(a, o, \{a_2\}) = [P_{identifier}(a) = P_{identifier}(a_1)] \wedge [P_{owner-id}(o) = P_{identifier}(a_2)] \wedge [P_{state}(a_2) = "busy"]$. The first condition indicates that this filter concerns the perception a given agent a_1 ; a_1 perceives the objects for which the property *owner* – *id* is defined and is equal to the *identifier* of a given agent a_2 iff the latter is busy. The observable properties of a special agent, the exchange of a message between two defined agents, the presence of a particular object or the combination of these instances can be used to define a particular context. A filter is the conjunction of at least two conditions, the first being related to the receptor and the second being related to o (in this case, there is no special context). This definition implies that the same o can be perceived according to different contexts, or on the contrary that, with the same context, several o can be perceived.

The whole system can have laws that cannot be broken e.g. a light signal cannot pass through an opaque object or an opaque fluid; the control by the environment is possible inside our model in a straightforward way [9]. In fact, the environment can put new filters itself and it manages the whole filters' scheduling; since its filters express the possibilities of interactions, they are executed before those of the agents. In order to express the environment's filters, we introduce the notion of *negative filters*. They are defined in the same way as filters, except that $f(a, o, C) \implies \neg perceive(a, o)$. Instead of enabling perception, the negative filters prevent it. As the environment may manage both filters and negative filters, it is possible to control standard behaviors, by enforcing and/or forbidding certain message transmissions or certain objects perception e.g. radius of perception in situated agents. The negative filters block any further filter that would concern the same a and o . Detailing these features is the purpose of the next section: we show how EASI can be used to supervise protocol executions and how it embeds the classical interaction patterns.

3 Protocol and Control

As we argue that the PBC principle embeds the other patterns of interaction, we demonstrate how to express them in a unified way via the EASI model. In this section, we introduce an example of the use of EASI, and we give the different possibilities a designer has thanks to the environment model. We consider a multiagent application in

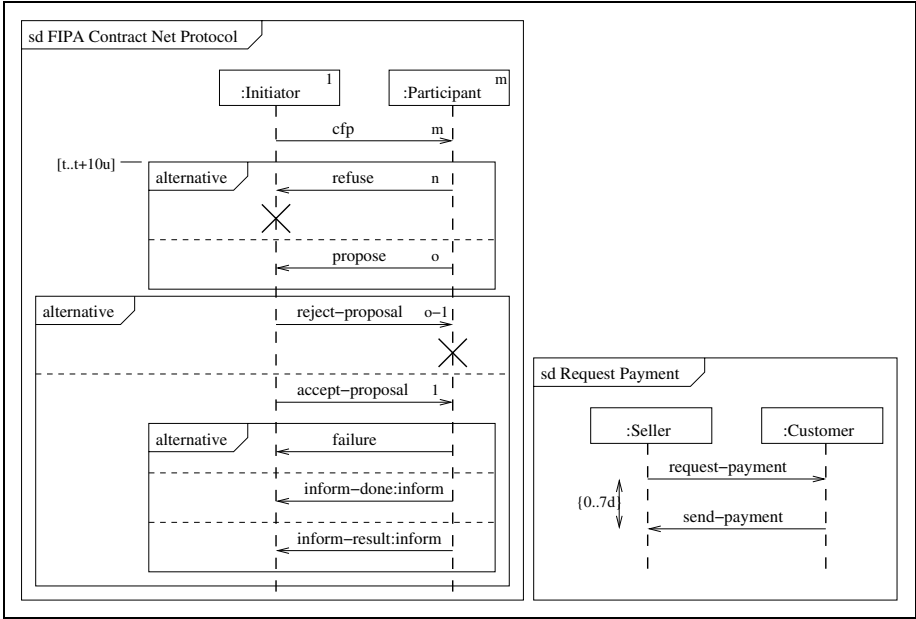


Fig. 2. FIPA Contract-Net Protocol (left) and Payment protocol (right)

the domain of electronic auctions. There are *seller* agents and *customer* agents. Every agent has two minimal properties: its “identifier” ($P_{identifier}$) and its “group” (P_{group}). In addition, the *customer* agents have a third property, “interest” ($P_{interest}$), which allows them to express what kind of item interests them, and the *seller* agents have a property “speciality” ($P_{speciality}$), which shows the kinds of goods they sell. The negotiation over the goods is fulfilled via a Contract-Net Protocol (CNP [4]) (cf. Fig. 2). The *customer* agent calls for proposals (cfp) for a good, and the *seller* may reject the cfp call, or make a proposition with a particular price. The buyer chooses one (or none) of the sellers and sends the confirmation or rejection messages to the corresponding agents. The messages are FIPA-ACL compliant, i.e. they exhibit as observable properties the parameters defined in FIPA-ACL¹: $P_{performative}$, $P_{conversation-id}$, P_{sender} etc. Finally, the *sender* sends the goods and requests the payment via a second protocol. In this MAS, we enable dyadic interaction via a first filter: every addressee of a message should receive it via $f_{Dyadic}(a, m) = [P_{identifier}(a) = P_{receiver}(m)]$. Depending on the policy of the MAS, the first message may be sent to every agent by broadcast, or to selected agents by multicast. The broadcast can be realized for every cfp message with the following filter: $f_{Broadcast\ cfp}(a, m, C) = [P_{identifier}(a) \neq null] \wedge [P_{performative}(m) = \text{“call-for-proposal”}]$. A multicast is achieved thanks to $f_{Multicast}(a, m) = [P_{identifier}(a) \in P_{receiver}(m)]$, considering that the property $P_{receiver}$ of the message is composed of several addresses.

¹ Foundation for Intelligent Physical Agents, Version J <http://www.fipa.org/specs/fipa00061/>

We may also integrate other models for particular messages. For example, the AGR (Agent, Group, Role) model proposes agents that are situated in groups and play roles [5] inside each group. If these two pieces of information appear as observable properties, it is possible to manage the interaction in the same way as the Madkit², which is the platform dedicated to this model. For instance, the sending of a message to the group of sellers is achieved thanks to $f_{Multicast\ Group}(a, m) = [P_{identifier}(a) \neq null] \wedge [P_{group} = "seller"]$. In the same way, it is possible to restrict broadcast to a role. If the initiator sends a message to the agents having a specific role in a group (a restricted broadcast) then the filter is: $f_{RG}(a, m, a1) = [P_{identifier}(a1) = P_{sender}(m)] \wedge [P_{group}(a1) \in P_{group}(a)] \wedge [P_{role}(a1) \in P_{role}(a)]$. This filter restricts the reception to messages which emitter belongs to one of the groups to which the receptor a belongs and who plays one of the roles that the receptor a plays.

Finally, we may explore *ad hoc* interactions, notably by taking advantage of the observability of the property "speciality". The buyer may send its message to the only sellers that sell a particular kind of goods. This may be achieved either at each message by putting the corresponding filter: $f_{Ad\ Hoc}(a, m) = [P_{speciality}(a) = spec] \wedge [P_{conversation-id} = id]$ with *spec* the speciality and *id* the particular id of the call for proposal it sends. Another way to achieve this is to enable a property "speciality" in the cfp messages; which is then managed by a generic filter each time this field is filled: $f_{Ad\ Hoc\ Gen}(a, m) = [P_{speciality}(a) = P_{speciality}(m)] \wedge [P_{performative}(m) = "call - for - proposal"]$. The remainder of the messages (proposals etc.) are sent via the dyadic filter, i.e. a classical dyadic interaction. However, the agents may overhear some messages that are not addressed to them, in order to monitor the MAS for example, or to improve their perception of their environment. Let the monitoring agent(s) exhibit $P_{group}(a) = "monitor"$. A non-discriminating monitoring filter would be:

$f_{Monitor}(a, m) = [P_{sender}(m) \neq null] \wedge [P_{group}(a) = "monitor"]$. If the agent wants to overhear only one agent a_1 , it modifies the first clause with $[P_{sender}(m) = P_{identifier}(a_1)]$; if it wants to overhear only acceptance messages it adds a clause $[P_{performative}(m) = "accept"]$. Other examples of focused overhearing are for a seller to receive the offers of the other sellers in order to be more competitive; or a buyer may want to receive offers it has not issued a cfp for. For the first situation, the filter is: $f_{awareness\ 1}(a, m, \{a_2\}) = [P_{identifier}(a) = P_{identifier}(a_1)] \wedge [P_{performative}(m) = "propose"] \wedge [P_{sender}(m) = P_{identifier}(a_2)] \wedge [P_{group}(a_2) = "seller"] \wedge [P_{speciality}(a_2) = "discs"]$: a_1 overhears all the "propose" messages issued by "discs" sellers. The filter that corresponds to the second situation is:

$f_{awareness\ 2}(a, m) = [P_{identifier}(a) = P_{identifier}(a_1)] \wedge [P_{performative}(m) = "propose"] \wedge [P_{speciality} = "books"]$: a_1 overhears every book proposal message. These examples describe how EASI enables the agents to achieve interactional awareness thanks to their filters and to specialize their perception.

When the negotiation is over, the seller asks the payment of the good(s) to the client (cf. Fig. 2). This protocol must not be overheard, as it may contain secured information. This is achieved by the two following negative filters, which secure the protocol by forbidding the reception of the messages with the performatives request-payment and send-payment by other agents than those specified as "receiver" parameter:

² <http://www.madkit.org>

$f_{OH\ Ban\ 1}(a, m, C) \implies \neg perceive(a, m) \text{ with } f_{OH\ Ban\ 1}(a, m, C) = (P_{identifier}(a) \neq P_{receiver}(m)) \wedge (P_{performative}(m) = \text{"request - payment"})$ and
 $f_{OH\ Ban\ 2}(a, m, C) \implies \neg perceive(a, m) \text{ with } f_{OH\ Ban\ 2}(a, m, C) = (P_{identifier}(a) \neq P_{receiver}(m)) \wedge (P_{performative}(m) = \text{"send - payment"})$. As the filter is managed by the environment, even the monitoring filters cannot overrule the perception ban. Note that the filter f_{Dyadic} automatically manages the messages for the intended receiver.

EASI enables the environment to support the MAS protocol management. In addition, thanks to the PBC principle, it enables the use, in the same MAS, of different interaction patterns such as AGR, multicast and indirect interactions. The designer employing EASI may choose for every situation the best solution, which can be dynamic during runtime, and equally use different means to adjust the protocol endorsement to the needs of the system.

4 Application: Agent Information Server

We apply the EASI model to an information server embedding agents representing both human users and transportation web services. The server informs users about transportation networks' status online. Every user has a specific goal during his connection to the server. The transportation web services exhibit their domain of expertise as observable properties. Transportation services providers can provide informations as a response to a request, or asynchronously by sending periodic notifications about disturbances, accidents, special events etc.

4.1 Technical Details

We implemented a web server, the environment is a rule-based engine wrapped inside the server, it handles rules (*filters*) and facts (both agents and objects) [12]. The only entities' attributes taken into account by the engine are observable properties. Every web service has a representative inside the server responsible of the convey of messages from the server to the physical port of the web service and inversely. Messages' exchange between the server and the web services are SOAP messages³ and asynchronous communication is fulfilled through the JAXM api⁴ for web services supporting SOAP, and a FTP server otherwise, used this way as a mailbox. This communication heterogeneity is of course transparent for the agents inside the environment i.e. they interact exactly the same way within the MAS environment whatever the transport protocol used is. Every user is physically mobile and connects via a MPTA (Mobile Personal Transport Assistant) to the server, and has during his session a representative agent inside the server, it is his interlocutor during his connection.

4.2 Execution

The problem in this kind of application domains concerns the information flows that are dynamic and asynchronous. Every information source is a hypothetical source of relevant

³ Simple Object Access Protocol <http://www.w3.org/TR/SOAP>

⁴ Java Api for XML Messaging, <http://java.sun.com/webservices/jaxm/>

information, thus an agent cannot know *a priori* which information it is interested in, it depends on its runtime changing context. Its context depends on a crossing of different information. Also, all the human users are not the same, their preferences are not homogeneous and their profiles have to be taken into account in their interaction with the system. Let us describe an example of use of the server. There is an abstraction of the transportation network inside the server, composed of stops. They are objects as described in the model. Every stop is described by a line number P_{line} to which it belongs and a position number P_{number} reflecting its position in the line. A user (its representative, say u) is described by its actual position in the network (a couple $(myline, mynumber)$), and by properties expressing its transportation modes' preferences (*mode*), the desired extra-transportation services (coffee-shops, parkings etc.). Basically, u has a path to follow during its trip i.e. a list of triples $(line, number_{source}, number_{destination})^+$, each triple represents a continuous trip, without change. u creates a filter f restricting its interaction to messages dealing with events occurring on its road. For instance, let $path(u) = [(2, 4, 10), (4, 5, 14)]$ reflecting that u has to change at the stop number 10, walk to stop number 5 of the line 4 and continue on the line 4 until the stop number 14. The corresponding filters are: $f_1(u, m) = [P_{id}(u) = myid] \wedge [P_{subject}(m) = \text{"alert"}] \wedge [P_{line}(m) = 2] \wedge [mynumber \leq P_{number}(m)] \wedge [P_{number}(m) \leq 10]$ and $f_2(u, m) = [P_{id}(u) = myid] \wedge [P_{subject}(m) = \text{"alert"}] \wedge [P_{line}(m) = 4] \wedge [5 \leq P_{number}(m)] \wedge [P_{number}(m) \leq 14]$. u is interested by the only alerts concerning its own path expressed in f_1 and f_2 . It is then notified about every alert event occurring in these segments of network. Since $mynumber$ is updated in every move, the concerned segment is reduced gradually until $mynumber = 10$, then u retracts f_1 and f_2 becomes: $f_2(u, m) = [P_{id}(u) = myid] \wedge [P_{subject}(m) = \text{"alert"}] \wedge [P_{line}(m) = 4] \wedge [mynumber \leq P_{number}(m)] \wedge [P_{number}(m) \leq 14]$ until the trip ends.

A private transportation operator, say p , has a representative in the server. It has the following filter: $f_{awareness}^p(a, m, a_1) = [P_{id}(a) = myid] \wedge [P_{subject}(m) = \text{"alert"}] \wedge [P_{line}(m) = P_{myline}(a_1)] \wedge [P_{mynumber}(a_1) \leq P_{number}(m)]$. The message that is caught by u is also caught by p , it knows that u has a disturbance on its road and can propose him an alternative transportation mode. p identifies this way a context in which its target market is involved, it identifies the subset of alert messages that are actually received by travelers and it uses this information to send addressed propositions to its possible customers. The described process is an application of the PBC model and it avoids the periodic requests for new relevant information, handles automatically *ad hoc* events occurring in the network and enables the agents to react following their runtime changing contexts.

5 Conclusion and Perspectives

We propose a generic coordination principle: the Property Based Coordination. Based on the notion of observability, it generalizes existing cooperation schemas as the Capability Based Cooperation, Agent Group Role, Overhearing, Mutual Awareness etc. We also presented the EASI model, a straightforward way to instantiate PBC. It enables the use of the environment as a sharing place where the agents manage their interaction according to their context. Entities composing the environment are agents and objects. The latter embeds the non-agent entities. With the observable properties, the entities

composing the environment can be clustered into agent categories, messages typology etc. which can be available as meta-information in the system. The environment is responsible of the aggregation of the filters to determine who perceives what (e.g. by a rule-based engine) and can restrict the interaction possibilities thanks to filters. In future works, we propose to investigate a formal model of actions in a system based on PBC and its dynamics. Applications from the transportation domain are implemented, in which we experiment new heuristics to solve the scheduling and routing problems, these heuristics are discovered thanks to the retained model of the environment.

References

1. H.-H. Bock and E. Diday. *Analysis of Symbolic Data. Exploratory Methods for Extracting Statistical Information from Complex Data*. Springer-Verlag, Heidelberg, second edition, 2000. 425 pages.
2. N. Carriero and D. Gelernter. *How to Write Parallel Programs: A First Course*. MIT press, Massachusetts, first edition, 1990. 250 pages.
3. N. Carriero, D. Gelernter, and J. Leichter. Distributed data structures in linda. In *Proceedings of the 13th ACM symposium on Principles of programming languages*, pages 236–242, Florida, USA, 1986.
4. R. Davis and R. G. Smith. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20:63–109, 1983.
5. J. Ferber, O. Gutknecht, C. Jonker, J. Muller, and J. Treur. Organization models and behavioural requirements specification for multi-agent systems. In *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS'00)*, pages 387–388, Boston, USA, 2000. IEEE.
6. J. Ferber and J. P. Müller. Influences and reaction: a model of situated multiagent systems. In *Proceedings of the second International Conference on Multi-Agent Systems (ICMAS'96)*, pages 72–79, Kyoto, Japan, 1996. AAAI Press.
7. E. Platon, N. Sabouret, and S. Honiden. Overhearing and direct interactions: Point of view of an active environment, a preliminary study. In D. Weyns, H. Parunak, and F. Michel, editors, *Environments For Multiagents Systems I*, volume 3374 of *Lecture Notes in Artificial Intelligence*, pages 121–138. Springer Verlag, 2005.
8. A. Ricci, M. Viroli, and A. Omicini. Programming mas with artifacts. In R. H. Bordini, M. Dastani, J. Dix, and A. E. Seghrouchni, editors, *Programming Multi-Agent Systems III*, volume 3862 of *Lecture Notes in Computer Science*, pages 206–221. Springer Verlag, 2006.
9. J. Saunier, F. Balbo, and F. Badeig. Environment as active support of interaction. In *Proceeding of the third workshop on Environment for Multiagent Systems (E4MAS'06)*, Hakodate, Japan, 2006. to appear.
10. K. Sycara, K. Decker, and M. Williamson. Middle-agents for the internet. In *Proceedings of the 15th Joint Conference on Artificial Intelligence (IJCAI'97)*, pages 578–583, 1997.
11. D. Weyns and T. Holvoet. A colored petri net for regional synchronization in situated multi-agent systems. In *Proceedings of First International Workshop on Petri Nets and Coordination*, Bologna, Italy, 2004. PNC.
12. M. Zargayouna, F. Balbo, and J. Saunier. Agent information server: a middleware for traveler information. In O. Dikenelli, M.-P. Gleizes, and A. Ricci, editors, *Engineering Societies in the Agents World VI*, volume 3963 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, 2006.

A Formal General Setting for Dialogue Protocols

Leila Amgoud, Sihem Belabbès, and Henri Prade

IRIT - CNRS, 118 route de Narbonne
31062 Toulouse Cedex 9, France
{amgoud, belabbes, prade}@irit.fr

Abstract. In this paper, we propose a general and abstract formal setting for argumentative dialogue protocols. We identify a minimal set of basic parameters that characterize dialogue protocols. By combining three parameters, namely the possibility or not of backtracking, the number of moves per turn and the turn-taking of the agents, we identify eight classes of protocols. We show that those classes can be reduced to three main classes: a ‘rigid’ class, an ‘intermediary’ one and a ‘flexible’ one. Although different proposals have been made for characterizing dialogue protocols, they usually take place in particular settings, where the locutions uttered and the commitments taken by the agents during dialogues and even the argumentation system that is involved are fixed. The present approach only assumes a minimal specification of the notion of dialogue essentially based on its external structure. This allows for protocol comparison and ensures the generality of the results.

Keywords: Protocol, Dialogue, Multi-agent systems.

1 Introduction

An important class of interactions between agents in multi-agent systems take the form of dialogues. There is a great variety of dialogues ranging from exchanges of pre-formatted messages to argumentation-based dialogues. In this latter category, Walton and Krabbe [1] distinguish six types of dialogue including negotiation, persuasion and information seeking. A key component for designing a dialogue system is its *protocol*. A protocol is a set of rules that govern the well-behaviour of interacting agents in order to generate dialogues. It specifies for instance the set of speech acts allowed in a dialogue and their allowed types of replies. A research trend views dialogues as dialogue games [2,3], where the agents are considered as playing a game with personal goals and a set of moves (i.e. instantiated speech acts) that can be used to try to reach those goals. Once a protocol has been fixed, choosing among moves is a strategy problem. While a protocol is a public notion independent of any mental state of the agents, a strategy is crucially an individualistic matter that refers to their personal attitude (being cooperative or not) and to their knowledge, in order to optimize their benefits w.r.t. their preferences.

Various dialogue protocols can be found in the literature, especially for persuasion [4,5] and negotiation [6,7,8,9,10,11,12] dialogues. A natural question then emerges about how to compare or categorize the existing dialogue protocols,

and more generally to characterize the minimal features that should be fixed for defining a protocol. This problem has been tackled in different ways. For instance in [13], dialogue protocols have been informally specified in terms of commencement, combination and termination rules. They have been compared essentially on the basis of the locutions uttered and the commitments taken by the agents during the generated dialogues. Subsequently and exploiting the same idea, dialogue protocols have been represented as objects of a category theory where the locutions and commitments are considered as morphisms [14].

In [15], a formal framework for persuasion dialogues have been proposed. The coherence of persuasion dialogues is ensured by relating a so-called ‘reply structure’ on the exchanged moves to the proof theory of argumentation theory [16]. This allows some flexibility on the structure of protocols regarding the turn-taking of the agents or the relevance of the moves.

The general setting that we propose in this paper can constitute the basis of further dialogue systems formalization. Namely, from a minimal set of basic parameters, eight classes of dialogue protocols are identified then further clustered in three main categories : a ‘rigid’ class, an ‘intermediary’ one and a ‘flexible’ one. This classification of protocols is essentially obtained by comparing the structure of the dialogues that they can generate, which in turn depends on the values of the basic parameters.

For instance, most of game-theoretic negotiation protocols such as bargaining [17], contract net [18] or e-commerce [19,20] are rather simple since agents only exchange offers and counter-offers¹. Thus they can be classified in the rigid class. On the contrary, protocols for argumentative dialogues [21,22] are more complex since agents not only put forward propositions, they also try to persuade one another about their validity through arguments. Arguments may be defeated thus it would be preferable to allow agents to try other argumentative tactics. Thus such protocols need more flexibility to be handled such as the ability of backtracking or playing several moves at the same turn.

Therefore, when dealing with some dialogue type, it would be enough to instantiate the proposed parameters, and to refer to a convenient protocol from one of the main identified classes. This would help to compare, for instance, negotiation approaches which is up to now undone.

This paper is organized as follows: Section 2 proposes the basic parameters that define a formal model for dialogue protocols. Section 3 studies the classes of protocols obtained by combining the parameters, and shows their relationships. Finally, Section 4 provides some discussions of related works w.r.t. the classes of protocols and concludes.

2 A Formal Setting for Dialogue Protocols

A protocol is a set of rules that govern the construction of dialogues between agents. Those rules come from fixing a set of *basic parameters* common to all

¹ Although those models focus on the design of appropriate strategies rather than complex protocols.

argumentative dialogue protocols. Different definitions of the parameters lead to distinct protocols that may generate structurally different dialogues.

We identify seven parameters considered as essential for defining any dialogue protocol, denoted by π , as a tuple $\pi = \langle \mathcal{L}, SA, Ag, Reply, Back, Turn, N_Move \rangle$ where:

1. \mathcal{L} is a *logical language*. Let $\text{Wff}(\mathcal{L})$ be the set of well-formed formulas of \mathcal{L} , and $\text{Arg}(\mathcal{L})$ the set of *arguments*² that can be built from \mathcal{L} .
2. SA is a set of *speech acts* or *locutions* uttered in a dialogue. Examples of speech acts are ‘*offer*’ for making propositions in a negotiation dialogue, ‘*assert*’ for making claims and ‘*argue*’ for arguing in a persuasion dialogue.
3. $Ag = \{a_1, \dots, a_n\}$ is a set of agents involved in a dialogue.
4. $\text{Reply} : SA \longrightarrow 2^{SA}$ is a function associating to each speech act its expected replies. For instance, a challenged claim needs to be replied to by an argument.
5. $\text{Back} \in \{0, 1\}$ is a variable such that $\text{Back} = 1$ (resp. 0) means that the protocol allows (resp. or not) for backtracking. This notion consists of replying to moves (i.e. speech acts with their contents) uttered at any earlier step of the dialogue, and not only to the previous one. If backtracking is forbidden, then a move is restricted to be a reply to the move uttered just before it. Backtracking may take two forms [15]:
 - *Alternative replies*: An agent may give some reply to a move and later in the dialogue it decides to change this reply by uttering an alternative one.
 - *Postponed replies*: An agent may delay its reply to some move to a later step of the dialogue because it prefers first replying to another move.
6. $\text{Turn} : \mathcal{T} \longrightarrow Ag$ is a function governing the turn-taking of the agents, where $\mathcal{T} = \{t_1, \dots, t_k, \dots \mid t_i \in \mathbb{N}, t_i < t_{i+1}\}$ is the set of turns taken by the agents. Most of existing protocols consider that the agents take turns during the generated dialogues. However, it is interesting to consider other turn-taking patterns:
 - *Take turns*: The turns shift uniformly to all the agents,
 - *Do not take turns*: The turns shift erratically, w.r.t. some given rules.
7. $\text{N_Move} : \mathcal{T} \times Ag \longrightarrow \mathbb{N}$ is a function determining at each turn and for each agent the number of moves that it is allowed to perform at that turn. It is defined as $\forall (t_i, a_j), \text{N_Move}(t_i, a_j) > 0$ iff $\text{Turn}(t_i) = a_j$. The opportunity of playing several moves per turn is well illustrated by argumentation-based negotiation dialogues. Indeed, an agent may propose an offer and arguments in its favour at the same turn in order to convince its peers [9].

Note that the above definition of Turn as a mapping from \mathcal{T} to Ag covers the special case where the agents take turns:

Proposition 1. *Let $Ag = \{a_1, \dots, a_n\}$. If $\forall t_i \in \mathcal{T}, \text{Turn}(t_i) = a_i \text{ modulo } n$, then the agents take turns (supposing without loss of generality that agent a_1 plays first, at turn t_1).*

² An argument is a reason to believe statements. Several definitions of arguments exist. See [23] for more details.

Similarly, the definition of **N_Move** includes the particular case where the agents perform exactly one move per turn:

Proposition 2. *If $\forall t_i \in \mathcal{T}, \forall a_j \in Ag, \text{Turn}(t_i) = a_j$ and $\text{N_Move}(t_i, a_j) = 1$, then the agents play exactly one move per turn.*

Protocols govern the construction of dialogues. Before defining that notion of dialogue, let us first introduce some basic concepts such as: *moves* and *dialogue moves*.

Definition 1 (Moves). *Let $\pi = \langle \mathcal{L}, SA, Ag, \text{Reply}, \text{Back}, \text{Turn}, \text{N_Move} \rangle$ be a protocol. A move m is a pair $m = (s, x)$ s.t. $s \in SA, x \in \text{Wff}(\mathcal{L})$ or $x \in \text{Arg}(\mathcal{L})$. Let \mathcal{M} be the set of moves that can be built from $\langle SA, \mathcal{L} \rangle$. The function **Speech** returns the speech act of the move m ($\text{Speech}(m) = s$), and **Content** returns its content ($\text{Content}(m) = x$).*

Some moves may not be allowed. For instance, a speech act ‘offer’ is usually used for exchanging offers in negotiation dialogues. Thus, sending an argument using this speech act is not a ‘well-formed’ move. This is captured by a mapping as follows:

Definition 2 (Well-formed moves). *Let $\text{WFM} : \mathcal{M} \longrightarrow \{0, 1\}$. A move $m \in \mathcal{M}$ is well-formed iff $\text{WFM}(m) = 1$.*

A second basic concept is that of ‘*dialogue move*’.

Definition 3 (Dialogue moves). *Let $\pi = \langle \mathcal{L}, SA, Ag, \text{Reply}, \text{Back}, \text{Turn}, \text{N_Move} \rangle$ be a protocol. A dialogue move M in the set of all dialogue moves denoted DM , is a tuple $\langle S, H, m, t \rangle$ such that:*

- $S \in Ag$ is the agent that utters the move, given by $\text{Speaker}(M) = S$
- $H \subseteq Ag$ denotes the set of agents to which the move is addressed, given by a function $\text{Hearer}(M) = H$
- $m \in \mathcal{M}$ is the move, given by a function $\text{Move}(M) = m$ and s.t. $\text{WFM}(m) = 1$
- $t \in DM$ is the target of the move i.e. the move which it replies to, given by a function $\text{Target}(M) = t$. We denote $t = \emptyset$ if M does not reply to any other move.

Dialogues are about *subjects* and aim at reaching *goals*. *Subjects* may take two forms w.r.t. the dialogue type.

Definition 4 (Dialogue subject). *A dialogue subject is φ such that $\varphi \in \text{Wff}(\mathcal{L})$, or $\varphi \in \text{Arg}(\mathcal{L})$.*

The *goal* of a dialogue is to assign a value to its subject pertaining to some domain. Two types of domains are distinguished according to the dialogue type.

Definition 5 (Dialogue goal). *The goal of a dialogue is to assign to its subject φ a value $v(\varphi)$ in a domain V such that:*

- If $\varphi \in \text{Wff}(\mathcal{L})$, then $v(\varphi) \in V = V_1 \times \dots \times V_m$
- If $\varphi \in \text{Arg}(\mathcal{L})$, then $v(\varphi) \in V = \{\text{acc}, \text{rej}, \text{und}\}$.

The nature of the domain V depends on the dialogue type. For instance, the subject of a negotiation dialogue with the goal of choosing a date and a place to organize a meeting, takes its values in $V_1 \times V_2$, where V_1 is a set of dates and V_2 a set of places. The subject of an inquiry dialogue with the goal of asking about the president's age, takes its values in a set V_1 of ages. Regarding persuasion dialogues whose goal is to assign an acceptability value to an argument³, the possible values are *acceptable*, *rejected* and *undecided*.

Let '?' denote an empty value. By default the subject of any dialogue takes this value.

Now that the basic concepts underlying the notion of a dialogue are introduced, let us define formally a *dialogue* conducted under a given protocol π .

Definition 6 (Dialogue). *Let $\pi = \langle \mathcal{L}, SA, Ag, Reply, Back, Turn, N_Move \rangle$ be a protocol. A dialogue d on a subject φ under the protocol π , is a non-empty (finite or infinite) sequence of dialogue moves, $d = M_{1,1}, \dots, M_{1,l_1}, \dots, M_{k,1}, \dots, M_{k,l_k}, \dots$ such that:*

1. $\varphi \in Wff(\mathcal{L})$ or $\varphi \in Arg(\mathcal{L})$. **Subject**(d) = φ returns the dialogue subject
2. $\forall M_{i,j}, i \geq 1, 1 \leq j \leq l_i, M_{i,j} \in DM$
3. $\forall i, i \geq 1, \mathbf{Speaker}(M_{i,j}) = \mathbf{Turn}(t_i)$
4. $\forall i, i \geq 1, l_i = \mathbf{N_Move}(t_i, \mathbf{Speaker}(M_{i,l_i}))$
5. $\forall i, i \geq 1, \mathbf{Speaker}(M_{i,1}) = \dots = \mathbf{Speaker}(M_{i,l_i})$
6. If $\mathbf{Target}(M_{i,j}) \neq \emptyset$,
then $\mathbf{Speech}(\mathbf{Move}(M_{i,j})) \in \mathbf{Reply}(\mathbf{Speech}(\mathbf{Move}(\mathbf{Target}(M_{i,j}))))$
7. $\forall j, 1 \leq j \leq l_1, \mathbf{Target}(M_{1,j}) = \emptyset$
8. $\forall M_{i,j}, i > 1, \mathbf{Target}(M_{i,j}) = M_{i',j'}$ such that:
 - If $\mathbf{Back} = 1$, then $1 \leq i' < i$ and $1 \leq j' \leq l_{i'}$
 - If $\mathbf{Back} = 0$, then $[(i - (n - 1)) \leq i' < i]$ and $1 \leq j' \leq l_{i'}$, where $[i - (n - 1)] \geq 1$ and n is the number of agents.

If the sequence $d = M_{1,1}, \dots, M_{1,l_1}, \dots, M_{k,1}, \dots, M_{k,l_k}, \dots$ is finite, then the dialogue d is *finite*, otherwise d is *infinite*.

We denote by \mathcal{D}_π the set of all dialogues built under the protocol π .

Condition 3 states that the speaker is defined by the function **Turn**. Condition 4 specifies the number of moves to be uttered by that agent. Condition 5 ensures that effectively that number of moves is uttered by that agent. Condition 6 ensures that the uttered speech act is a legal reply to its target. Condition 7 states that the initial moves played at the first turn by the agent which starts the dialogue do not reply to any other move. Condition 8 regulates backtracking for all moves different from the initial ones. Indeed, if backtracking is allowed then a move can reply to any other move played previously in the dialogue. Otherwise, this is restricted to the moves played by the other agents at their last turn just before the current one.

³ Dung [16] has defined three semantics for the *acceptability* of arguments. An argument can be accepted, rejected or in abeyance. Formal definitions of those status of arguments are beyond the scope of this paper.

This notion of non-backtracking can be illustrated as follows: consider a dialogue between two agents that take turns to play exactly one move per turn. If backtracking is forbidden, then each move (except the first one) replies to the one played just before it.

The above definition of backtracking captures its two types: an *alternative* reply and a *postponed* reply.

Definition 7. Let $\pi = \langle \mathcal{L}, SA, Ag, Reply, Back, Turn, N_Move \rangle$ be a protocol. Let $d \in \mathcal{D}_\pi$ with $d = M_{1,1}, \dots, M_{1,l_1}, \dots, M_{k,1}, \dots, M_{k,l_k}, \dots$. Let $M_{i,j}, M_{i',j'} \in DM$ s.t. $Target(M_{i,j}) = M_{i',j'}$. If $Back = 1$, then:

- $M_{i,j}$ is an ‘alternative reply’ to $M_{i',j'}$ iff $\exists f, i' < f < i$ and $\exists k, 1 \leq k \leq l_f$ s.t. $Target(M_{f,k}) = M_{i',j'}$ and $Speaker(M_{f,k}) = Speaker(M_{i,j})$
- $M_{i,j}$ is a ‘postponed reply’ to $M_{i',j'}$ iff $\forall f, i' < f < i$ and $\forall k, 1 \leq k \leq l_f$, s.t. if $Target(M_{f,k}) = M_{i',j'}$ then $Speaker(M_{f,k}) \neq Speaker(M_{i,j})$.

Each dialogue has an *outcome* which represents the value assigned to its subject. This *outcome* is given by a function as follows:

Definition 8 (Dialogue outcome). Let $\pi = \langle \mathcal{L}, SA, Ag, Reply, Back, Turn, N_Move \rangle$ be a protocol. $Outcome : \mathcal{D}_\pi \rightarrow V \cup \{?\}$, s.t. $Outcome(d) = ?$ or $Outcome(d) = v(Subject(d))$.

Note that if d is infinite, then $v(Subject(d)) = ?$ thus $Outcome(d) = ?$.

The outcome of a dialogue is not necessarily an optimal one. In order to compute the optimal outcome of a dialogue, it is necessary to specify its type, to fix all the parameters of the protocol that generates it (such as its set of speech acts), and also to specify the belief and goals bases of the agents. This is beyond the scope of this paper.

3 Classes of Dialogue Protocols

In this section, we combine three basic binary parameters, namely: *backtracking*, *turn-taking* and *number of moves per turn*. This leads to eight classes of protocols that are then compared on the basis of the structure of dialogues that they can generate. We show that some classes are equivalent, and that others are less rigid than others w.r.t. the dialogue structure. Two types of results are presented: those valid for dialogues between multiple agents ($n > 2$), and those that hold only for two agents dialogues.

In order to compare classes of protocols, we need to compare pairs of dialogues that they generate. Johnson *et al.* [13] have discussed how to determine when two dialogue protocols are similar. Their approach is based on syntax (e.g. speech acts) or agents’ commitments. Our view is more semantically oriented in that we consider a notion of *equivalent dialogues* based on their subject and the outcome that they reach, by insuring that they have some moves in common. Formally:

Definition 9 (Equivalent dialogues). Let $\pi_1 = \langle \mathcal{L}, SA, Ag, Reply, Back, Turn, N_Move \rangle$ and $\pi_2 = \langle \mathcal{L}, SA, Ag, Reply, Back, Turn, N_Move \rangle$ be two protocols. Let $d_1 \in \mathcal{D}_{\pi_1}$ and $d_2 \in \mathcal{D}_{\pi_2}$ be two finite dialogues. Let DM_1 and DM_2

denote the set of dialogue moves of d_1 and d_2 respectively. d_1 is equivalent to d_2 , denoted $d_1 \sim d_2$, iff: i) $\text{Subject}(d_1) \equiv^4 \text{Subject}(d_2)$, ii) $\text{Outcome}(d_1) = \text{Outcome}(d_2)$, and iii) $DM_1 \cap DM_2 \neq \emptyset$.

Let us take an illustrative example.

Example 1. The following dialogues between a_1 and a_2 are equivalent.⁵

a_1 : Offer(x)	a_1 : Offer(x)
a_2 : Argue(S, x), (where $S \vdash x$)	a_2 : Refuse(x)
a_1 : Accept(x)	a_1 : Why_refuse(x)?
	a_2 : Argue($S', \neg x$), (where $S' \vdash \neg x$)
	a_1 : Argue(S, x), (where $S \vdash x$)
	a_2 : Accept(x)

We define *equivalent protocols* as protocols that generate equivalent dialogues.

Definition 10 (Equivalent protocols). Let $\pi_1 = \langle \mathcal{L}, SA, Ag, Reply, Back, Turn, N_Move \rangle$ and $\pi_2 = \langle \mathcal{L}, SA, Ag, Reply, Back, Turn, N_Move \rangle$ be two protocols. π_1 is equivalent to π_2 , denoted $\pi_1 \approx \pi_2$, iff $\forall d_1 \in \mathcal{D}_{\pi_1}, \exists d_2 \in \mathcal{D}_{\pi_2}$ s.t. $d_1 \sim d_2$, and $\forall d_2 \in \mathcal{D}_{\pi_2}, \exists d_1 \in \mathcal{D}_{\pi_1}$ s.t. $d_2 \sim d_1$.

In all what follows, Π denotes a class of protocols. If any dialogue conducted under Π_1 has an equivalent dialogue under Π_2 , then $\mathcal{D}_{\Pi_1} \subseteq \mathcal{D}_{\Pi_2}$, and we write $\Pi_1 \subseteq \Pi_2$.

Before comparing the eight classes of protocols obtained by combining the aforementioned parameters, the following results can be established where simplified notations are adopted for the values of the parameters:

- $x = \bar{B}$ if **Back** = 0, $x = B$ if **Back** = 1,
- $y = T$ if **Turn** requires taking turns, $y = \bar{T}$ otherwise,
- $z = S$ if **N_Move** allows for single move per turn, $z = M$ for multiple moves.

Let Π be a class of protocols. Π_{xyz} stands for the class of protocols such that, everything being equal elsewhere, the parameters **Back**, **Turn** and **N_Move** take respectively the values x, y and z . Note that we only index the parameters whose values are modified.

The following result shows that a class of protocols where one parameter is assigned some value is included in the class of protocols where this parameter takes the opposite value.

Proposition 3. Let Π_x, Π_y and Π_z be three classes of protocols (where x, y and z are defined as above). The following inclusions hold: i) $\Pi_{\bar{B}} \subseteq \Pi_B$, ii) $\Pi_T \subseteq \Pi_{\bar{T}}$, and iii) $\Pi_S \subseteq \Pi_M$.

Then, combinations of pairs of parameters lead to the following inclusions:

⁴ \equiv stands for logical equivalence.

⁵ The role of each speech act in both dialogues can be easily understood from its designation.

Proposition 4. Let Π_{xz} , Π_{yz} and Π_{xy} be three classes of protocols. We have: i) $\Pi_{BS} \subseteq \Pi_{BM}$, ii) $\Pi_{TS} \subseteq \Pi_{TM}$, and iii) $\Pi_{BT} \subseteq \Pi_{BT}$.

The next result shows that combining pairs of parameters gives birth to equivalent classes of protocols. If $n > 2$, then we can only state the inclusion relationship between classes where the parameter *turn-taking* is fixed.

Proposition 5. Let Π_{xz} , Π_{yz} and Π_{xy} be three classes of protocols. The following equivalences hold:

- $\Pi_{BM} \approx \Pi_{BS}$
- If $n = 2$, then $\Pi_{TM} \approx \Pi_{TS}$. If $n > 2$, then $\Pi_{TM} \subseteq \Pi_{TS}$
- If $n = 2$, then $\Pi_{BT} \approx \Pi_{BT}$. If $n > 2$, then $\Pi_{BT} \subseteq \Pi_{BT}$.

Finally, it is worth pointing out that by fixing the three parameters, we are able to compare the eight classes of protocols and to identify the equivalent ones.

Proposition 6. Let Π_{xyz} be a class of protocols. The following equivalences and inclusions hold:

- If $n = 2$, then
 $\Pi_{BTS} \subseteq \Pi_{BTM} \approx \Pi_{BTS} \approx \Pi_{BTM} \approx \Pi_{BTS} \approx \Pi_{BTM} \approx \Pi_{BTS} \subseteq \Pi_{BTM}$
- If $n > 2$, then
 $\Pi_{BTS} \subseteq \Pi_{BTS} \approx \Pi_{BTM} \subseteq \Pi_{BTM} \subseteq \Pi_{BTS} \approx \Pi_{BTM} \subseteq \Pi_{BTM}$, and
 $\Pi_{BTS} \subseteq \Pi_{BTS} \approx \Pi_{BTM} \subseteq \Pi_{BTM} \subseteq \Pi_{BTS} \approx \Pi_{BTM} \subseteq \Pi_{BTM}$.

This result shows that when dealing with interactions between two agents, the eight classes of protocols reduce to three classes. Thus, a protocol for any dialogue system involving two agents can be formalized by choosing the adequate protocol in one of those three classes.

It also shows the intuitive result that protocols of the class Π_{BTS} , i.e. generating dialogues where backtracking is forbidden, the agents take turns and play one move per turn, are the most ‘*rigid*’ ones in terms of dialogue structure. This gathers for instance e-commerce protocols. Conversely, protocols of the class Π_{BTM} , i.e. generating dialogues where backtracking is allowed, the agents do not take turns and play several moves per turn, are the most ‘*flexible*’ ones. This encompasses for instance argumentation-based dialogue protocols. Indeed, most of game-theoretic negotiation protocols such as bargaining [17], contract net [18] or e-commerce [19,20] are rather simple since agents only exchange offers and counter-offers⁶. Thus they can be classified in the rigid class. On the contrary, protocols for argumentative dialogues [21,22] are more complex and need more flexibility to be handled. The remaining protocols are called ‘intermediary’ in that they allow for flexibility on one or two parameters among the three binary ones, but not on all of them at the same time. For instance, protocols from the class Π_{BTS} impose some rigidity by enforcing the agents to play a single move per turn.

⁶ Although those models focus on the design of appropriate strategies rather than complex protocols.

4 Discussion and Conclusion

In this paper, we have proposed a general and abstract framework for dialogue protocols. In particular, we have presented a minimal set of basic parameters that define a protocol. Some parameters depend on the intended application of the framework. Indeed, the logical language used in the framework, the set of speech acts and the replying function directly relate to the dialogue type and to the context in which it occurs. Other parameters are more generic since they relate to the external structure of the generated dialogues. Those are the possibility or not of backtracking, the turn-taking of the agents and the number of moves per turn. Combinations of those three parameters give birth to eight classes of protocols.

In the particular case of dialogues between two agents, and which is the most common in the literature, we have shown that those classes reduce to three main classes: a first class containing ‘rigid’ protocols, a second one containing ‘intermediary’ ones, and a third one containing ‘flexible’ ones. We have also studied the relationships between the eight classes in the general case of dialogues between more than two agents.

Recently, Prakken [15] has proposed a formal dialogue system for persuasion, where two agents aim at resolving a conflict of opinion. Each agent gives arguments in favour of its opinion or against the one of its opponent, such that arguments conflict. By analyzing the defeat relation between those arguments and counterarguments, the introduced protocol is structured as an argument game, like the Dung’s argumentation proof theory [16]. Thus each performed move is considered to ‘attack’ or ‘surrender to’ a previous one, and is attributed a ‘dialogical status’ as either ‘in’ or ‘out’. A labeling procedure governs the assignment of those statuses. It allows for defining a turn-taking rule, it regulates backtracking by checking the relevance of moves. This framework is intended to maintain coherence of persuasion dialogues. Although it is well-defined and well-motivated, it is specific to dialogue systems modeling defeasible reasoning. With respect to our main result, this protocol belongs to the intermediary class denoted by Π_{BTM} , where $n = 2$ (the number of agents).

We now consider other protocols that can be found in the literature. For instance, McBurney *et al.* [22] have proposed an informal protocol for deliberation dialogues between more than two agents. Agents interact to decide what course of action should be adopted in a given situation. Following this protocol, agents do not take turns, utter single move per turn and are not allowed to backtrack. Thus it is contained in the intermediary class $\Pi_{\bar{B}TS}$ where $n > 2$.

In a game-theoretic negotiation context, Alonso [6] has introduced a protocol for task allocation. Dialogues consist of sequences of offers and counter-offers between two agents. Agents take turns and utter single move per turn. Backtracking is allowed but not formalized. The protocol is in the intermediary class Π_{BTS} where $n = 2$.

In an e-commerce scenario, Fatima *et al.* [19] have proposed a negotiation protocol where two agents (a ‘buyer’ and a ‘seller’) bargain over the price of an item. Agents alternately exchange multi-attribute bids (or offers) in order to

reach an acceptable one. This protocol can then be classified in the most rigid class Π_{BTS} where $n = 2$.

Thus we believe that in order to formalize a dialogue system, it would be enough to instantiate the minimal basic parameters w.r.t. the constraints of the domain or the application, and to refer to a convenient protocol in one of the main identified classes.

This classification being based on the external structure of dialogues, we are currently working on the coherence of dialogues. Indeed, the different protocols that we have proposed can generate incoherent dialogues in the sense that we do not impose conditions on when the performed moves are allowed. We plan to get rid of this by examining each dialogue type with its own specificities, and more importantly by considering agents' mental states. In other words, we need to look at agents' strategies to obtain a complete dialogue framework.

We are also examining a list of suitable quality criteria for evaluating dialogue protocols, such as the capacity to reach an outcome and the efficiency in terms of the quality of the outcome. Some of those criteria depend directly on the parameters for backtracking, the number of moves per turn and the turn-taking, while others relate to particular instantiations of the framework. Of course the proposed parameters are not exhaustive so we intend to identify a wider variety of them such as those relating to agents' roles in a dialogue or to dialogue execution time. Evaluation criteria for game-theoretic negotiation protocols [24,25] already exist. However, as they are related to agents' individual utilities which remain static in such contexts, they could not be used for dialogues where agents' preferences may evolve through dialogue. A tentative of transposing game-theoretic criteria to argumentative dialogues has been proposed [26] but they are informal.

To sum up, such development may help to identify classes of protocols that are more suitable for each dialogue type, and also to evaluate them. For instance, negotiation dialogues can be conducted under flexible or rigid protocols, depending on whether the approach allows or not for arguing. We would then be able to compare negotiation approaches, namely: game-theoretic, heuristic-based and argumentation-based ones.

References

1. D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, Albany, 1995.
2. C.L. Hamblin. *Fallacies*. Methuen, London, 1970.
3. J. MacKenzie. Question-begging in non-cumulative systems. *Journal of Philosophical Logic*, 8:117–133, 1979.
4. S. Parsons, M. Wooldridge, and L. Amgoud. On the outcomes of formal inter-agent dialogues. In *Proc. AAMAS'03*, pages 616–623, Melbourne, 2003.
5. H. Prakken. Relating protocols for dynamic dispute with logics for defeasible argumentation. *Synthese*, 127:187–219, 2001.
6. E. Alonso. A formal framework for the representation of negotiation protocols. *Inteligencia Artificial*, 3/97:30–49, 1997.
7. L. Amgoud, S. Parsons, and N. Maudet. Arguments, dialogue, and negotiation. In *Proc. ECAI'00*, pages 338–342, Berlin, 2000.

8. M. K. Chang and C. C. Woo. A speech-act-based negotiation protocol: design, implementation, and test use. *ACM Transactions on Information Systems*, 12(4):360–382, 1994.
9. S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, 104(1–2):1–69, 1998.
10. S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.
11. F. Sadri, F. Toni, and P. Torroni. An abductive logic programming architecture for negotiating agents. In *Proc. JELIA'02*, LNAI 2424, pages 419–431, Cosenza, 2002.
12. C. Sierra, N. R. Jennings, P. Noriega, and S. Parsons. A framework for argumentation-based negotiation. In *Proc. 4th Intl. Workshop on Agent Theories, Architectures and Languages (ATAL'97)*, pages 167–182, Providence, 1997.
13. M. Johnson, P. McBurney, and S. Parsons. When are two protocols the same? In *Communication in Multiagent Systems: Agent Communication Languages and Conversation Policies*, LNAI 2650, pages 253–268, 2003.
14. M. Johnson, P. McBurney, and S. Parsons. A mathematical model of dialog. *Electronic Notes in Theoretical Computer Science*, 2005. In press.
15. H. Prakken. Coherence and flexibility in dialogue games for argumentation. *Journal of Logic and Computation*, 15:1009–1040, 2005.
16. P. M. Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n -person games. *Artificial Intelligence*, 77:321–357, 1995.
17. A. Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50:97–109, 1982.
18. R. G. Smith. The contract net protocol: high-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, 29(12):1104–1113, 1980.
19. S.S. Fatima, M. Wooldridge, and N. R. Jennings. An agenda based framework for multi-issues negotiation. *Artificial Intelligence*, 152(1):1–45, 2004.
20. A.R. Lomuscio, M. Wooldridge, and N.R. Jennings. A classification scheme for negotiation in electronic commerce. *Intl. Journal of Group Decision and Negotiation*, 12(1):31–56, 2003.
21. J. van Veenen and H. Prakken. A protocol for arguing about rejections in negotiation. In *Proc. 2nd Intl Workshop on Argumentation in Multi-Agent Systems (ArgMas-05)*, Utrecht, 2005.
22. P. McBurney, D. Hitchcock, and S. Parsons. The eight-fold way of deliberation dialogue. *Intelligent Systems*, 2005. In press.
23. H. Prakken and G. Vreeswijk. Logics for defeasible argumentation. In *Handbook of Philosophical Logic, second edition*, volume 4, pages 218–319. 2002.
24. J.S. Rosenschein and G. Zlotkin. *Rules of encounter: Designing conventions for automated negotiation among computers*. MIT Press, Cambridge, 1994.
25. T.W. Sandholm. Distributed rational decision making. In G. Wei, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 201–258. MIT Press, Cambridge, 1999.
26. P. McBurney, S. Parsons, and M. Wooldridge. Desiderata for agent argumentation protocols. In *Proc. AAMAS'02*, pages 402–409, Bologna, 2002.

OCC's Emotions: A Formalization in a BDI Logic^{*}

Carole Adam, Benoit Gaudou, Andreas Herzig, and Dominique Longin

Université Paul Sabatier – IRIT/LILaC

118 route de Narbonne, F-31062 Toulouse CEDEX 9 (France)

{adam, gaudou, herzig, longin}@irit.fr

Abstract. Nowadays, more and more artificial agents integrate emotional abilities, for different purposes: expressivity, adaptability, believability... Designers mainly use Ortony et al.'s typology of emotions, that provides a formalization of twenty-two emotions based on psychological theories. But most of them restrain their agents to a few emotions among these twenty-two ones, and are more or less faithful to their definition. In this paper we propose to extend standard BDI (belief, desire, intention) logics to account for more emotions while trying to respect their definitions as exactly as possible.

1 Introduction

Recently, the agent community gets very interested in emotional artificial agents with enhanced abilities including expressivity, adaptability and believability. To cope with the increasing complexity of such agents, designers need rigorous formal models offering properties like genericity and verifiability.

Current computer models of emotions are mainly semi-formal or only manage a limited number of emotions, and are thus often specific to the context of use. However, Meyer [1] proposes a very formal model of four emotions, but as he states himself [1, p.11], his goal was not to “capture the informal psychological descriptions exactly (or as exact as possible)” but rather to describe what “makes sense for artificial agents”. In this paper we provide a logical formalization of twenty emotions while staying as close as possible to one of the most cited psychological approaches, *viz.* that of Ortony, Clore, and Collins (OCC) [2]. Compared to the previous version of this work, we manage twelve more emotions, and we have modified the existing ones to be more faithful to OCC. These emotions are formalized inside a BDI modal logic (Belief, Desire, Intention), that has numerous advantages: widespread thus broadly studied, established results of verifiability and genericity, strong explicative power of the agent's behavior... Our architecture grounds on previous work [3]. We here omit the notions of choice and intention (that turned out to be avoidable), and add a probabilistic belief operator, as well as a refined notion of desire with its symmetric notion of “undesire”, and deontic operators.

There exist several kinds of models of emotions. Discrete models (*e.g.* [5,6]) are mainly descriptive. Dimensional models (*e.g.* [7]) are practical when aiming at describing the dynamics and expression of emotions (*e.g.* [8]) but not explicative of the triggering of emotions. Finally, cognitive models (*e.g.* [2,9,10]) assume that emotions are

^{*} A preliminary version of this work has been published in the ECAI workshop AITaMI'06. The authors would like to thank the AIMSA reviewers for their very useful comments.

triggered by the cognitive evaluation of stimuli following some judgement criteria or *appraisal variables*. They are more normative than other models, and thus we can find them as a basis in many intelligent agent architectures. Some rare researchers prefer the complex theories from Lazarus (e.g. [11]) or Frijda (e.g. [12]), but most of them (e.g. [13]), including this paper, ground on the model of Ortony, Clore, and Collins (OCC).

The OCC typology has three branches, each of which corresponds to the appraisal of a different type of stimulus with respect to a particular appraisal variable, and related to particular mental attitudes. For example, the stimulus event “it is raining” is appraised as being undesirable w.r.t. the agent’s goal of taking coffee on a terrace. These branches are then differentiated into several groups of emotion types with similar eliciting conditions.

Section 2 introduces our logical framework allowing to express the necessary intensity variables. Sections 3 and 4 detail the event-based and agent-based branches of the typology.

2 Logical Framework

Our formal framework is based on the modal logic of belief, choice, time, and action of Herzig and Longin [3] which is a refinement of Cohen and Levesque’s works [14]. We need neither choice nor intention (build from belief and choice), thus we do not use them. We extend this logic with modal probability operators defined by Herzig [4], as well as obligation and desirability operators.

Semantics. Let AGT be the set of agents, ACT the set of actions, $ATM = \{p, q, \dots\}$ the set of atomic formulas. The set of complex formulas will be noted $FORM = \{\varphi, \psi, \dots\}$. A possible-worlds semantics is used, and a model \mathcal{M} is a triple $\langle W, V, \mathcal{R} \rangle$ where W is a set of possible worlds, V is a truth assignment which associates each world w with the set V_w of atomic propositions true in w , and \mathcal{R} is a tuple of structures made up of:

- $\mathcal{A} : ACT \rightarrow (W \rightarrow 2^W)$ which associates each action $\alpha \in ACT$ and possible world $w \in W$ with the set $\mathcal{A}_\alpha(w)$ of possible worlds resulting from the execution of action α in w ;
- $\mathcal{B} : AGT \rightarrow (W \rightarrow 2^W)$ which associates each agent $i \in AGT$ and possible world $w \in W$ with the set $\mathcal{B}_i(w)$ of possible worlds compatible with the beliefs of agent i in w . All these accessibility relations are serial;
- $\mathcal{P} : AGT \rightarrow (W \rightarrow 2^{2^W})$ which associates each agent $i \in AGT$ and possible world $w \in W$ with a set of sets of possible worlds $\mathcal{P}_i(w)$. Intuitively for $U \in \mathcal{P}_i(w)$, U contains more elements than its complement $\mathcal{B}_i \setminus U$;
- $\mathcal{G} : W \rightarrow 2^W$ which associates each possible world $w \in W$ with the set $\mathcal{G}(w)$ of possible worlds in the future of w . This relation is a linear order (reflexive, transitive and antisymmetric). $\mathcal{G} \supseteq \mathcal{A}_\alpha$ for every α ;
- $\mathcal{L} : AGT \rightarrow (W \rightarrow 2^W)$ (resp. $\mathcal{D} : AGT \rightarrow (W \rightarrow 2^W)$) which associates each agent $i \in AGT$ and possible world $w \in W$ with the set $\mathcal{L}_i(w)$ (resp. $\mathcal{D}_i(w)$) of possible worlds compatible with what the agent i likes (resp. dislikes) in the world w . All these accessibility relations are serial. Moreover, for the sake of simplicity, we make the simplistic hypothesis that what is liked persists: if $w \mathcal{G} w'$ then $\mathcal{L}_i(w) = \mathcal{L}_i(w')$. Similarly for what is disliked;

- $\mathcal{I} : AGT \rightarrow (W \rightarrow 2^W)$ which associates each agent $i \in AGT$ and possible world $w \in W$ with the set $\mathcal{I}_i(w)$ of ideal worlds for the agent i . In these ideal worlds all the (social, legal, moral...) obligations, norms, standards... of agent i hold. All these relations are serial.¹

We associate modal operators to these mappings: $After_\alpha \varphi$ reads “ φ is true after every execution of action α ”, $Before_\alpha \varphi$ reads “ φ is true before every execution of action α ”, $Bel_i \varphi$ reads “agent i believes that φ ”, $Prob_i \varphi$ reads “for i φ is more probable than $\neg\varphi$ ”, $G\varphi$ reads “henceforth φ is true”, $H\varphi$ reads “ φ has always been true in the past”, $Idl_i \varphi$ reads “ideally it is the case for i that φ ” and $Des_i \varphi$ (resp. $Undes_i \varphi$) reads “ φ is desirable (resp. undesirable) for i ”.

The truth conditions are standard for almost all of our operators: $w \Vdash \Box\varphi$ iff $w' \Vdash \varphi$ for every $w' \in \mathcal{R}_\Box(w)$ where $\mathcal{R}_\Box \in \mathcal{A} \cup \mathcal{B} \cup \{\mathcal{G}\} \cup \mathcal{I}$ and $\Box \in \{After_\alpha : \alpha \in ACT\} \cup \{Bel_i : i \in AGT\} \cup \{G\} \cup \{Idl_i : i \in AGT\}$ respectively. For the converse operators we have: $w \Vdash \Box\varphi$ iff $w' \Vdash \varphi$ for every w' such that $w \in \mathcal{R}_\Box(w')$ where $\mathcal{R}_\Box \in \mathcal{A} \cup \{\mathcal{G}\}$ and $\Box \in \{Before_\alpha : \alpha \in ACT\} \cup \{H\}$ respectively. Furthermore, $w \Vdash Prob_i \varphi$ iff there is $U \in \mathcal{P}_i(w)$ such that for every $w' \in U$, $w' \Vdash \varphi$.

Intuitively, φ is desirable for agent i if i likes φ and does not dislike φ , viz. φ is true in every world i likes and is false in at least one world i dislikes: $w \Vdash Des_i \varphi$ iff for every $w' \in \mathcal{L}_i(w)$, $w' \Vdash \varphi$ and there is a world $w'' \in \mathcal{D}_i(w)$ such that $w'' \not\Vdash \varphi$. In a similar way: $w \Vdash Undes_i \varphi$ iff for every $w' \in \mathcal{D}_i(w)$, $w' \Vdash \varphi$ and there is a world $w'' \in \mathcal{L}_i(w) : w'' \not\Vdash \varphi$. It follows from these constraints that φ cannot be simultaneously desirable and undesirable, and that there are φ that are neither desirable nor undesirable (e.g. tautologies and inconsistencies).

We have the following introspection constraints: if $w \in \mathcal{B}_i(w')$ then $\mathcal{B}_i(w) = \mathcal{B}_i(w')$, $\mathcal{P}_i(w) = \mathcal{P}_i(w')$, $\mathcal{L}_i(w) = \mathcal{L}_i(w')$ and $\mathcal{D}_i(w) = \mathcal{D}_i(w')$, insuring that agents are aware of their beliefs, probabilities, desires, and “undesires”. We also require that $U \subseteq \mathcal{B}_i(w)$ for every $U \in \mathcal{P}_i(w)$, ensuring that belief implies probability.

Dynamic Operators. $After_\alpha$ and $Before_\alpha$ are defined in the standard tense logic K_t , viz. logic K with conversion axioms (see [16] for more details). For every α and φ , as $\mathcal{G} \supseteq \mathcal{A}_\alpha$, we have that $G\varphi \rightarrow After_\alpha \varphi$. As we suppose that time is linear, $Happens_\alpha \varphi \stackrel{def}{=} \neg After_\alpha \neg\varphi$ reads “ α is about to happen, after which φ ” and $Done_\alpha \varphi \stackrel{def}{=} \neg Before_\alpha \neg\varphi$ reads “ α has just been done, and φ was true before”.

In the following, the notation $i:\alpha$ reads “agent i is the author of action α ”.

Belief Operators. Bel_i operators are defined in the standard KD45 logic that we do not develop here (see [17,15] for more details).

Temporal Operators. The logic of G and H is linear temporal logic with conversion axioms (see [16] for more details). $F\varphi \stackrel{def}{=} \neg G\neg\varphi$ reads “ φ is true or will be true at some future instant”. $P\varphi \stackrel{def}{=} \neg H\neg\varphi$ reads “ φ is or was true”.

Probability Operators. The probability operators correspond to a notion of weak belief. It is based on the notion of subjective probability measure. The logic of $Prob$ is much

¹ We disregard thus conflicts between different kinds of standards.

weaker than the one of belief, in particular it is non-normal: the necessitation rule and the axiom K of belief operators do not have any counterpart in terms of *Prob*.

Belief and Probability. They are related by the validity of:

$$Bel_i \varphi \rightarrow Prob_i \varphi \quad (\text{BPR})$$

We define an abbreviation $Expect_i \varphi$, reading “*i* believes that φ is probably true, but envisages the possibility that it could be false”.

$$Expect_i \varphi \stackrel{\text{def}}{=} Prob_i \varphi \wedge \neg Bel_i \varphi \quad (\text{Def}_{Expect})$$

Desirable/Undesirable Operators. They represent preference in a wide sense. We here consider that an agent finds φ desirable, undesirable or φ leaves him indifferent. Due to the truth condition for Des_i , the following principles are valid:

$$\begin{array}{llll} \neg Des_i \perp & (\perp_{Des_i}) \\ \text{if } \varphi \leftrightarrow \psi \text{ then } Des_i \varphi \leftrightarrow Des_i \psi & (\text{RE}_{Des}) & Des_i \varphi \rightarrow GDes_i \varphi & (\text{Pers}_{Des_i}) \\ Des_i \varphi \rightarrow \neg Des_i \neg \varphi & (\text{D}_{Des}) & \neg Des_i \varphi \rightarrow G\neg Des_i \varphi & (\text{Pers}_{\neg Des_i}) \\ \neg Des_i \top & (\top_{Des_i}) & Des_i \varphi \rightarrow \neg Undes_i \varphi & (\text{RDU}) \end{array}$$

(Pers_{Des_i}) and $(\text{Pers}_{\neg Des_i})$ illustrate that what is desirable is atemporal. We also have corresponding principles for $Undes_i$. Note that desires are neither closed under implication nor under conjunction: I might desire to marry Ann and to marry Beth without desiring to be a bigamist. Finally, (RDU) expresses that something cannot be desirable and undesirable at the same time.

Obligation Operator. The notion of obligation considered here is very wide: it embraces all the rules agents ideally have to respect. They can be explicit (like laws) or more or less implicit (like social or moral obligations). They are a kind of social preferences, imposed by a group to which the agent pertains, and thus differ from the agent's personal desires. The logic of *Idl* is the standard deontic logic KD (thus an agent's “obligations” must be consistent).

We will now formalize Ortony et al.'s emotions: we cite OCC's informal definition, give a formal definition, and illustrate it by OCC's examples.

3 Event-Based Emotions

The event-based branch of the OCC typology contains emotion types whose eliciting conditions depend on the evaluation of an event, with respect to the agent's goals. *Desirability* is a central intensity variable accounting for the impact that an event has on an agent's goals, *i.e.* how it helps or impedes their achievement. We formalize it through our *Des* and *Undes* operators.

3.1 Well-Being Emotions

The emotion types in this group have eliciting conditions focused on the desirability for the self of an event. An agent feels joy (resp. distress) when he is pleased (resp. displeased) about a desirable (resp. undesirable) event.

$$\begin{aligned} Joy_i \varphi &\stackrel{\text{def}}{=} Bel_i \varphi \wedge Des_i \varphi \\ Distress_i \varphi &\stackrel{\text{def}}{=} Bel_i \varphi \wedge Undes_i \varphi \end{aligned}$$

For example in [2, p. 88]², when a man i hears that he inherits of a small amount of money from a remote and unknown relative k ($Bel_i(earn\text{-}money \wedge k\text{-}died)$), he feels **joy** because he focuses on the desirable event ($Des_i earn\text{-}money$). Though, this man does not feel distress about his relative's death, because this is not undesirable for him ($\neg Undes_i k\text{-}died$). On the contrary, a man j (p. 89) who runs out of gas on the freeway ($Bel_j out\text{-}of\text{-}gas$) feels **distress** because this is undesirable for him ($Undes_j out\text{-}of\text{-}gas$).

3.2 Prospect-Based Emotions

The emotion types in this group have eliciting conditions focused on the desirability for self of an anticipated (uncertain) event, that is actively prospected. They use a local intensity variable, *likelihood*, accounting for the expected probability of the event to occur. We formalize this variable with the operator *Expect*.

An agent feels hope (resp. fear) if he is “pleased (resp. displeased) about the **prospect** of a desirable (resp. undesirable) event”³.

$$\begin{aligned} Hope_i \varphi &\stackrel{\text{def}}{=} Expect_i \varphi \wedge Des_i \varphi \\ Fear_i \varphi &\stackrel{\text{def}}{=} Expect_i \varphi \wedge Undes_i \varphi \end{aligned}$$

The agent feels fear-confirmed (resp. satisfaction) if he is “displeased (resp. pleased) about the **confirmation** of the prospect of an undesirable (resp. desirable) event”.

$$\begin{aligned} FearConfirmed_i \varphi &\stackrel{\text{def}}{=} Bel_i PExpect_i \varphi \wedge Undes_i \varphi \wedge Bel_i \varphi \\ Satisfaction_i \varphi &\stackrel{\text{def}}{=} Bel_i PExpect_i \varphi \wedge Des_i \varphi \wedge Bel_i \varphi \end{aligned}$$

The agent feels relief (resp. disappointment) if he is “pleased (resp. displeased) about the **disconfirmation** of the prospect of an undesirable (resp. desirable) event”.

$$\begin{aligned} Relief_i \varphi &\stackrel{\text{def}}{=} Bel_i PExpect_i \neg \varphi \wedge Undes_i \neg \varphi \wedge Bel_i \varphi \\ Disappointment_i \varphi &\stackrel{\text{def}}{=} Bel_i PExpect_i \neg \varphi \wedge Des_i \neg \varphi \wedge Bel_i \varphi \end{aligned}$$

For example a woman w who applies for a job (p. 111) might feel **fear** if she expects not to be offered the job ($Expect_w \neg get\text{-}job$), or feel **hope** if she expects that she will be offered it ($Expect_w get\text{-}job$). Then, if she hoped to get the job and finally gets it, she feels **satisfaction**; and if she does not get it, she feels **disappointment**. An employee e (p. 113) who expects to be fired ($Expect_e f$) will feel **fear** if it is undesirable for him ($Undes_e f$), but not if he already envisaged to quit this job ($\neg Undes_e f$). In the first case he will feel **relief** when he is not fired ($Bel_e \neg f$), and **fear-confirmed** when he is.

² Below, the quoted pages all refer to OCC's book [2] so we just specify it once.

³ Note that the object of hope is not necessarily about the future: I might ignore whether my email has been delivered to the addressee, and hope it has.

Theorem. We can prove some links between emotions: $Satisfaction_i \varphi \rightarrow Joy_i \varphi$ and $FearConfirmed_i \varphi \rightarrow Distress_i \varphi$. This is in agreement with Ortony et al.'s definitions. Though, we can notice that the disconfirmation-centered emotions (relief and disappointment) do not imply the corresponding well-being emotions (joy and sadness). This seems rather intuitive, since they typically do not characterize a desirable or undesirable situation, but the return to an indifferent situation that was expected to change and that finally did not.

3.3 Fortunes-of-Others Emotions

The emotion types in this group have eliciting conditions focused on the presumed desirability for another agent. They use three local intensity variables: *desirability for other*, *deservingness*, and *liking*. *Desirability for other* is the assessment of how much the event is desirable for the other one: for example we write $Bel_i Des_j \varphi$ for “agent i believes that φ is desirable for agent j ”. *Deservingness* represents how much agent i believes that agent j deserved what occurred to him. It often depends on *liking*, i.e. i 's attitude towards j . Below, to simplify, we assimilate “ i believes that j deserves A ” and “ i desires that j believes A ”. We thus only consider *liking*, through non-logical global axioms. For example, when John likes Mary this means that if John believes that Mary desires to be rich, then John desires that Mary is rich, or rather: gets to know that she is rich ($Bel_{john} Des_{mary} rich \rightarrow Des_{john} Bel_{mary} rich$).

There are two good-will (or empathetic) emotions: an agent feels happy for (resp. sorry for) another agent if he is pleased (resp. displeased) about an event presumed to be desirable (resp. undesirable) for this agent.

$$\begin{aligned} HappyFor_{i,j} \varphi &\stackrel{def}{=} Bel_i \varphi \wedge Bel_i F Bel_j \varphi \wedge Bel_i Des_j \varphi \wedge Des_i Bel_j \varphi \\ SorryFor_{i,j} \varphi &\stackrel{def}{=} Bel_i \varphi \wedge Bel_i F Bel_j \varphi \wedge Bel_i Undes_j \varphi \wedge Undes_i Bel_j \varphi \end{aligned}$$

There are two ill-will emotions: an agent feels resentment (resp. gloating) towards another agent if he is displeased (resp. pleased) about an event presumed to be desirable (resp. undesirable) for this agent.

$$\begin{aligned} Resentment_{i,j} \varphi &\stackrel{def}{=} Bel_i \varphi \wedge Bel_i F Bel_j \varphi \wedge Bel_i Des_j \varphi \wedge Undes_i Bel_j \varphi \\ Gloating_{i,j} \varphi &\stackrel{def}{=} Bel_i \varphi \wedge Bel_i F Bel_j \varphi \wedge Bel_i Undes_j \varphi \wedge Des_i Bel_j \varphi \end{aligned}$$

For example (p. 95) Fred feels **happy for** Mary when she wins a thousand dollars, because he has an interest in the happiness and well-being of his friends (global axiom: $Bel_f Des_m w \rightarrow Des_f Bel_m w$). A man i (p. 95) can feel **sorry for** the victims v of a natural disaster ($Bel_i Bel_v disaster \wedge Bel_i Undes_v disaster$) without even knowing them, because he has an interest that people do not suffer undeservedly ($Undes_i Bel_v disaster$). An employee e (p. 99) can feel **resentment** towards a colleague c who receives a large pay raise ($Bel_e pr, Bel_e Des_c pr$) because he thinks this colleague is incompetent and thus does not deserve this raise ($Undes_e Bel_c pr$). Finally, Nixon's political opponents (p. 104) might have felt **gloating** about his departure from office ($Bel_o Bel_{nixon} d, Bel_o Undes_{nixon} d$) because they thought it was deserved ($Des_o Bel_{nixon} d$).

4 Agent-Based Emotions

The agent-based branch of the OCC typology contains emotion types whose eliciting conditions depend on the judgement of the praiseworthiness of an action, with respect to standards. An action is *praiseworthy* (resp. *blameworthy*) when it upholds (resp. violates) standards. We represent standards through the deontic operator *Idl*.

4.1 Attribution Emotions

The emotion types in this group have eliciting conditions focused on the approving of an agent's action. They use two local intensity variables: *strength of unit*⁴ and *expectation deviation*. *Expectation deviation* accounts for the degree to which the performed action differs from what is usually expected from the agent, according to his social role or category⁵. We express this with the formula $\neg Prob_i Happens_{j:\alpha} \top$, reading “*i* does not believe that it is likely that *j* performs successfully action α ”. Then $Done_{i:\alpha} \neg Prob_i Happens_{i:\alpha} \top$ expresses that surprisingly for himself, *i* succeeded in executing α . In the sequel, $Emotion_i(i:\alpha)$ abbreviates $Emotion_i Done_{i:\alpha} \top$ where *Emotion* is the name of an emotion.

Self-agent emotions: an agent feels pride (resp. shame) if he is approving (resp. disapproving) of his own praiseworthy (resp. blameworthy) action.

$$Pride_i(i:\alpha) \stackrel{def}{=} Bel_i Done_{i:\alpha} (\neg Prob_i Happens_{i:\alpha} \top \wedge Bel_i Idl_i Happens_{i:\alpha} \top)$$

$$Shame_i(i:\alpha) \stackrel{def}{=} Bel_i Done_{i:\alpha} (\neg Prob_i Happens_{i:\alpha} \top \wedge Bel_i Idl_i \neg Happens_{i:\alpha} \top)$$

Emotions involving another agent: an agent feels admiration (resp. reproach) towards another agent if he is approving (resp. disapproving) of this agent's praiseworthy (resp. blameworthy) action.

$$Admiration_{i,j}(j:\alpha) \stackrel{def}{=} Bel_i Done_{j:\alpha} (\neg Prob_i Happens_{j:\alpha} \top \wedge Bel_i Idl_j Happens_{j:\alpha} \top)$$

$$Reproach_{i,j}(j:\alpha) \stackrel{def}{=} Bel_i Done_{j:\alpha} (\neg Prob_i Happens_{j:\alpha} \top \wedge Bel_i Idl_j \neg Happens_{j:\alpha} \top)$$

For example, a woman *m* feels **pride** (p. 137) of having saved the life of a drowning child ($Bel_m Done_{m:\alpha} \top$, where α is the action to save the child) because she thinks that her action is praiseworthy, *i.e.* its successful execution was not expected (before α , it held that $\neg Prob_m Happens_{m:\alpha} \top$) but ideally she had to perform it ($Bel_m Idl_m Happens_{m:\alpha} \top$). A rich elegant lady *l* (p. 142) would feel **shame** if caught while stealing clothes in an exclusive boutique ($Bel_l Done_{l:\beta} \top$, where β is the action to steal), because this violates a standard ($Idl_l \neg Happens_{l:\beta} \top$) and this was not expected of herself ($\neg Prob_l Happens_{l:\beta} \top$). A physicist *p*'s colleagues *c* (p. 145) feel **admiration** towards him for his Nobel-prize-winning work ($Bel_c Done_{p:\gamma} \top$, where

⁴ *Strength of unit* intervenes in self-agent emotions to represent the degree to which the agent identifies himself with the author of the action, allowing him to feel pride or shame when he is not directly the actor. For example one can be proud of his son succeeding in a difficult examination, or of his rugby team winning the championship. In this paper we only focus on emotions felt by the agent about his own actions, thus we do not represent this variable.

⁵ In self-agent emotions the agent refers to the stereotyped representation he has of himself.

γ is the action to make some Nobel-prize-winning findings) because this is praiseworthy, *i.e.* ideal ($Bel_c Idl_p Happens_{p:\alpha} \top$) but very unexpected ($\neg Prob_c Happens_{p:\gamma} \top$). A man i may feel **reproach** towards a driver j (p. 145) who drives without a valid license ($Bel_i Done_{j:\delta} \top$, where δ is the action to drive without a valid license), because it is forbidden ($Bel_i Idl_j \neg Happens_{j:\delta} \top$), and it is not expected from a driver ($\neg Prob_i Happens_{j:\delta} \top$).

Theorem. We can prove that $Admiration_{i,i}(\varphi) \leftrightarrow Pride_i(\varphi)$ and $Reproach_{i,i}(\varphi) \leftrightarrow Shame_i(\varphi)$. This is rather intuitive, all the more Ortony et al. introduce the term *self-reproach* for shame.

4.2 Composed Emotions

These emotions occur when the agent focuses on both the consequences⁶ of the event and its agency. They are thus the result of a combination of well-being emotions and attribution emotions.

$$Gratification_i(i:\alpha, \varphi) \stackrel{def}{=} Pride_i(i:\alpha) \wedge Bel_i Before_{i:\alpha} \neg Bel_i F\varphi \wedge Joy_i\varphi$$

$$Remorse_i(i:\alpha, \varphi) \stackrel{def}{=} Shame_i(i:\alpha) \wedge Bel_i Before_{i:\alpha} \neg Bel_i F\varphi \wedge Distress_i\varphi$$

$$Gratitude_{i,j}(j:\alpha, \varphi) \stackrel{def}{=} Admiration_{i,j}(j:\alpha) \wedge Bel_i Before_{j:\alpha} \neg Bel_i F\varphi \wedge Joy_i\varphi$$

$$Anger_{i,j}(j:\alpha, \varphi) \stackrel{def}{=} Reproach_{i,j}(j:\alpha) \wedge Bel_i Before_{j:\alpha} \neg Bel_i F\varphi \wedge Distress_i\varphi$$

For example, a woman i may feel **gratitude** (p. 148) towards the stranger j who saved her child from drowning ($Bel_i Done_{j:\alpha} \top$, where α is the action to save her child). Indeed, she feels admiration towards j because of j 's praiseworthy action (*i.e.* ideal: $Bel_i Idl_j Happens_{j:\alpha} \top$, but unlikely: $\neg Prob_i Happens_{j:\alpha} \top$, for example because it needs a lot of courage). Moreover the effect of j 's action ($Bel_i son\text{-}alive$) is desirable for her ($Des_i son\text{-}alive$), so she also feels joy about it ($Joy_i son\text{-}alive$). Similarly, a woman w (p. 148) may feel **anger** towards her husband h who forgets to buy the groceries ($Bel_w Done_{h:\beta} \top$, where β is his action to come back without groceries), because the result of this action ($Bel_w \neg g$) is undesirable for her ($Undes_w \neg g$), and the action was blameworthy ($\neg Prob_w Happens_{h:\beta} \top \wedge Bel_w Idl_h Happens_{h:\beta} \top$). The physicist p may feel **gratification** about winning the Nobel prize because his action γ was praiseworthy, and its result ($Bel_p is\text{-}nobel$ would have been false if p had not performed γ) is desirable for him ($Des_p is\text{-}nobel$). Finally, a spy may feel **remorse** (p. 148) about having betrayed his country (action ω) if he moreover caused undesirable damages ($Shame_{spy}(\omega) \wedge Distress_{spy} damages \wedge Bel_{spy} Before_{spy:\omega} \neg Bel_{spy} F damages$).

It follows from our logic, in particular from the introspection axioms for all operators, that $Emotion_i\varphi \leftrightarrow Bel_i Emotion_i\varphi$ and $\neg Emotion_i\varphi \leftrightarrow Bel_i \neg Emotion_i\varphi$ are valid.

5 Conclusion

We have formalized twenty emotions from Ortony et al.'s theory (all but the object-based branch), thus providing a very complete set of emotions. Moreover we have

⁶ Here, we represent the effects of an action α with the formula $Bel_i Before_{i:\alpha} \neg Bel_i F\varphi$ reading approximately “ i believes that φ would not have been true if he had not performed α ”.

shown the soundness of our framework by illustrating each definition by an example from their book. We have privileged richness, genericity, and fidelity to the definitions over tractability. An optimization would have needed important concessions. For example [18] proposes a numerical model of emotions in combat games, efficient in big real-time multi-agent systems, but domain-dependant.

We would like to highlight here some shortcomings of our model. Mainly, our emotions are not quantitative (they have no intensity). This prevents us from fine-grained differentiations among emotions of the same type (for example: irritation, anger, rage). A second (and linked) shortcoming is that our emotions are persistent as long as their conditions stay true. Thereby some emotions (like *Joy* or *Satisfaction*) can persist *ad vitam eternam*, which is not intuitive at all. Indeed it is psychologically grounded that after an emotion is triggered, its intensity decreases, and when it is under a threshold, the emotion disappears. Finally, we cannot manage emotional blending of several emotions that are simultaneously triggered; [19] proposes an original solution to this issue. On our part, we leave these problems for further work.

References

1. Meyer, J.J.: Reasoning about emotional agents. In Proceedings of ECAI'04 (2004) 129–133
2. Ortony, A., Clore, G., Collins, A.: The cognitive structure of emotions. CUP (1988)
3. Herzig, A., Longin, D.: C&L intention revisited. In Proceedings of KR'04 (2004) 527–535
4. Herzig, A.: Modal probability, belief, and actions. *Fundamenta Informaticæ* **57**(2-4) (2003) 323–344
5. Darwin, C.R.: The expression of emotions in man and animals. Murray, London (1872)
6. Ekman, P.: An argument for basic emotions. *Cognition and Emotion* **6** (1992) 169–200
7. Russell, J.A.: How shall an emotion be called? In Plutchik, R., Conte, H., eds.: *Circumplex models of personality and emotions*. APA, Washington, DC (1997) 205–220
8. Becker, C., Kopp, S., Wachsmuth, I.: Simulating the emotion dynamics of a multimodal conversational agent. In: ADS'04, Springer LNCS (2004)
9. Lazarus, R.S.: *Emotion and Adaptation*. Oxford University Press (1991)
10. Frijda, N.H.: *The emotions*. Cambridge University Press, Cambridge, UK (1986)
11. Gratch, J., Marsella, S.: A domain-independent framework for modeling emotion. *Journal of Cognitive Systems Research* **5**(4) (2004) 269–306
12. Staller, A., Petta, P.: Introducing emotions into the computational study of social norms: a first evaluation. *Journal of artificial societies and social simulation* **4**(1) (2001)
13. Jaques, P.A., Vicari, R.M., Pesty, S., Bonneville, J.F.: Applying affective tactics for a better learning. In Proceedings of ECAI'04, IOS Press (2004)
14. Cohen, P.R., Levesque, H.J.: Intention is choice with commitment. *Artificial Intelligence Journal* **42**(2–3) (1990) 213–261
15. Chellas, B.F.: *Modal Logic: an introduction*. Cambridge University Press (1980)
16. Burgess, J.P.: Basic tense logic. In Gabbay, D., Guenther, F., eds.: *Handbook of Philosophical Logic*. Volume VII. Second edn. Kluwer Academic Publishers (2002)
17. Hintikka, J.: *Knowledge and Belief: An Introduction to the Logic of the Two Notions*. Cornell University Press, Ithaca (1962)
18. Parunak, H., Bisson, R., Brueckner, S., Matthews, R., Sauter, J.: A model of emotions for situated agents. In Stone, P., Weiss, G., eds.: *AAMAS'06*, ACM Press (2006) 993–995
19. Gershenson, C.: Modelling emotions with multidimensional logic. In: *NAFIPS'99*. (1999)

A Boolean Encoding Including SAT and n-ary CSPs

Lionel Paris, Belaïd Benhamou, and Pierre Siegel

Université de Provence,
LSIS - UMR CNRS 6168,
Marseille, France

{Belaïd.Benhamou, Lionel.Paris, Pierre.Siegel}@cmi.univ-mrs.fr

Abstract. We investigate in this work a generalization of the known *CNF* representation which allows an efficient Boolean encoding for n-ary CSPs. We show that the space complexity of the Boolean encoding is identical to the one of the classical CSP representation and introduce a new inference rule whose application until saturation achieves arc-consistency in a linear time complexity for n-ary CSPs expressed in the Boolean encoding. Two enumerative methods for the Boolean encoding are studied: the first one (equivalent to MAC in CSPs) maintains full arc-consistency on each node of the search tree while the second (equivalent to FC in CSPs) performs partial arc-consistency on each node. Both methods are experimented and compared on some instances of the Ramsey problem and randomly generated 3-ary CSPs and promising results are obtained.

Keywords: Logic and constraint programming, Automated reasoning, Knowledge representation and reasoning.

1 Introduction

Constraint solving is a well known framework in artificial intelligence. Mainly, two approaches are well used: the propositional calculus holding the satisfiability problem (SAT) and the formalism of discrete constraint satisfaction problems (CSPs).

Methods to solve the SAT problem in propositional calculus are former than the CSP ones. They have been widely studied for many years since the Davis and Putnam (DP for abbreviation) procedure [4] was introduced, and are still a domain of investigation of a large community of researchers. Several improvements of the DP method have been provided recently [12,9,5]. These new methods are able to solve large scale of SAT instances and are used to tackle real applications. The advantage of SAT methods is their flexibility to solve any constraints given in the CNF form. They are not sensitive to constraint arity¹ as it is often the case for CSP methods. One drawback of the CNF formulation is the loss of the problem structure which is well represented in CSP formalism.

The discrete CSP formalism was introduced by Montanari in 1974 [10]. The asset of this formalism is its capability to express the problem and explicitly represent its structure as a graph or a hyper-graph. This structure helps to achieve constraint propagation and to provide heuristics which render CSP resolution methods efficient. In comparison to the propositional calculus, the CSP resolution methods are sensitive to constraint

¹ Every SAT solveur can deal with any clause length with the same efficiency, except for binary clause.

arity. Most of the known methods [6] apply on binary² CSPs. This makes a considerable restriction since most of the real problems (see CSPLib³) need constraints of unspecified arity for their natural formulation. To circumvent this restriction problem, some works provide methods to deal with more general constraints [3], but this approach is still not well investigated.

Both SAT and CSP formulations are closely linked. Several works on transformations of binary CSP to SAT forms exist [8]. The transformation in [7] is generalized to non-binary CSPs in [2]. Transformations of a SAT form to a CSP form are described in [1,13]. Unfortunately, most of the transformations from CSP to SAT result in an overhead of space and lose the problem structure. Both factors combined may slow the resolution of the problem. Our aim in this paper is to provide a more general Boolean representation which includes both the CNF and the CSP formulations and which preserves their advantages. That is, a representation which does not increase the size of the problem and which keeps the problem structure. We show particularly how non-binary CSPs are well expressed in this new formulation and efficiently solved. We propose two enumerative methods for the general Boolean formulation which are based on the DP procedure. To enforce constraint propagation we implement a new inference rule which takes advantage of the encoded structure. The application of this rule to the considered problem is achieved with a linear time complexity. We will show that the saturation of this rule on the considered problem is equivalent to enforcing arc consistency on the corresponding CSP. We proved good time complexity to achieve arc consistency for non-binary CSPs. This allows to maintain efficiently full arc consistency at each node of the search tree. This makes the basis of the first enumerative method which is equivalent to the known method MAC [11] in CSP. We also prove that a partial exploitation of this inference rule leads to a second enumerative method which is equivalent to a forward checking (FC) method for non-binary CSPs. Authors of [3] showed that FC is not easy to be generalized for n -ary CSPs, they obtain six versions. In our encoding, FC is obtained naturally by applying the inference rule to the neighborhood of the current variable under instantiation.

The rest of the paper is organized as follows: first we recall some background on both propositional calculus and discrete CSP formalisms. Then we introduce a general normal form which we use to express both SAT and CSP problems. After that, we define a new inference rule which we use to show results on arc consistency. We describe two enumerative algorithms (the FC and MAC versions) based on two different uses of the introduced inference rule. Next, we give some experiments on both randomly generated non-binary CSPs and Ramsey problems to show and compare the behaviors of our resolution methods. Finally, we summarize some previous related works and conclude the work.

2 Background

A CNF formula f in propositional logic is a conjunction $f = C_1 \wedge C_2 \dots C_k$ of clauses. Each clause C_i is itself a disjunction of literals. That is, $C_i = l_1 \vee l_2 \dots l_m$ where each

² A binary CSP contains only two-arity constraints.

³ <http://csplib.org>

literal l_i is an occurrence of a Boolean variable either in its positive or negative parity. An interpretation I is a mapping which assigns to each Boolean variable the value *true* or *false*. A clause is satisfied if at least one of its literals l_i is given the value *true* in the interpretation I ($I[l_i] = \text{true}$). The empty clause is unsatisfiable and is denoted by \square . The formula f is satisfied by I if all its clauses are satisfied by I , thus I is a *model* of f . A formula is satisfiable if it admits at least one model, otherwise it is unsatisfiable.

On the other hand a CSP is a statement $P = (X, D, C, R)$ where $X = \{X_1, X_2, \dots, X_n\}$ is a set of n variables, $D = \{D_1, D_2, \dots, D_n\}$ is a set of finite domains where D_i is the domain of possible values for X_i , $C = \{C_1, C_2, \dots, C_m\}$ is a set of m constraints, where the constraint C_i is defined on a subset of variables $\{X_{i_1}, X_{i_2}, \dots, X_{i_{a_i}}\} \subset X$. The arity of the constraint C_i is a_i and $R = \{R_1, R_2, \dots, R_m\}$ is a set of m relations, where R_i is the relation corresponding to the constraint C_i . R_i contains the permitted combinations of values for the variables involved in the constraint C_i . A binary CSP is a CSP whose constraints are all of arity two (binary constraints). A CSP is non-binary if it involves at least a constraint whose arity is greater than 2 (a n-ary constraint). An *instantiation* I is a mapping which assigns each variables X_i a value of its domain D_i . A constraint C_i is satisfied by the instantiation I if the projection of I on the variables involved in C_i is a tuple of R_i . An instantiation I of a CSP P is consistent (or called a solution of P) if it satisfies all the constraints of P . A CSP P is consistent if it admits at least one solution. Otherwise P is not consistent. Both propositional satisfiability (SAT) and constraint satisfaction problems (CSPs) are two closely related NP-complete problems. For the sequel we denote by n the number of variables of the CSP, by m its number of constraints, by a its maximal constraint arity and by d the size of its largest domain.

3 An Encoding Including SAT and CSPs

The idea of translating a CSPs into an equivalent SAT form was first introduced by De Kleer in [8]. He proposed the well known *direct encoding* and since that Kasif [7] proposed the *AC encoding* for binary CSPs. More recently Bessière et al [2] generalized the *AC encoding* to non-binary CSPs. Our approach is different, it consists in providing a general Boolean form including both CNF (SAT) and CSP representations, rather than translating CSPs into SAT forms. We describe in this section, a new Boolean encoding which generalizes the CNF formulation, and show how n-ary CSPs are naturally represented in an optimal way (no overhead in size in comparison to the CSP representation) in this encoding.

3.1 The Generalized Normal Form (GNF)

A generalized clause C is a disjunction of Boolean formulas $f_1 \vee \dots \vee f_m$ where each f_i is a conjunction of literals, i.e $f_i = l_1 \wedge l_2 \wedge \dots \wedge l_n$. A formula is in Generalized Normal Form (GNF) if and only if it is a conjunction of generalized clauses. The semantic of generalized clauses is trivial: the generalized clause C is satisfied by an interpretation I if at least one of its conjunctions f_i ($i \in [1, m]$) is given the value *true* in I , otherwise it is falsified by I . A classical clause is a simplified generalized clause where all the

conjunctions f_i are reduced to single literals. This proves that GNF is a generalization of CNF. We show in the sequel that each constraint C_i of a given n -ary CSP is represented by a generalized clause. We reach the optimal size representation by using the cardinality formulas $(\pm 1, L)$ which means "exactly one literal among those of the list L have to be assigned the value *true* in each model", to express efficiently that a CSP variable has to be assigned a single value in its domain. We denote by *CGNF* the *GNF* augmented by the cardinality.

3.2 The CGNF Encoding for n -ary CSPs

Given an n -ary CSP $P = (X, D, C, R)$, first, we define the set of Boolean variables which we use in the Boolean representation, and two types of clauses: the domain clauses and the constraint clauses necessary to encode the domains and respectively the constraints.

- The set of Boolean variables: as in the existing encodings [8,7,2] we associate a Boolean variable Y_v with each possible value v of the domain of each variable Y of the CSP. Thus, $Y_v = \text{true}$ means that the value v is assigned to the variables Y of the CSP. We need exactly $\sum_{i=1}^n |D_i|$ Boolean variables. The number of Boolean variables is bounded by nd .
- The domain clauses: let Y be a CSP variable and $D_Y = \{v_0, v_1, \dots, v_k\}$ its domain. The cardinality formula $(\pm 1, Y_{v_0} \dots Y_{v_k})$ forces the variable Y to be assigned to only one value in D_Y . We need n cardinality formulas to encode the n variable domains.
- The constraint clauses: each constraint C_i of the CSP P is represented by a generalized clause C_{C_i} defined as follows: Let R_i be the relation corresponding to the constraint C_i involving the set of variables $\{X_{i_1}, X_{i_2}, \dots, X_{i_a}\}$. Each tuple $t_j = (v_{j_1}, v_{j_2}, \dots, v_{j_a})$ of R_i is expressed by the conjunction $f_j = X_{v_{j_1}} \wedge X_{v_{j_2}} \wedge \dots \wedge X_{v_{j_a}}$. If R_i contains k tuples t_1, t_2, \dots, t_k then we introduce the generalized clause $C_{C_i} = f_1 \vee f_2 \vee \dots \vee f_k$ to express the constraint C_i . We need m generalized clauses to express the m constraints of the CSP.

As the domain clauses are encoded in $O(nd)$, the constraint clauses in $O(mad^a)$, then the *CGNF* encoding of a CSP P is in $O(mad^a + nd)$ in the worst case. In fact, it is in $O(mad^a)$ since nd is often negligible. This space complexity is identical to the one of the original CSP encoding. This justifies the optimality in space of the *CGNF* encoding. Authors in [2] gave a spatial complexity in $O(mad^a)$ in the worst case for the k -AC encoding. As far as we understand, this complexity is rounded and does not take into account neither the at-least-one nor the at-most-one clauses. This increases the complexity to $O(mad^a + nd^2)$.

Definition 1. Let P be a CSP and C its corresponding *CGNF* encoding. Let I be an interpretation of C . We define the corresponding equivalent instantiation I_p in the CSP P as the instantiation verifying the following condition: for all CSP variable X and each value v of its domain, $X = v$ if and only if $I[X_v] = \text{true}$.

Theorem 1. Let P be a CSP, C its *CGNF* corresponding encoding, I an interpretation of the *CGNF* encoding and I_p the corresponding equivalent instantiation of I in the CSP P . I is a model of C if and only if I_p is a solution of the CSP P .

Proof. For a lack of space, the proof is omitted.

The *CGNF* encoding allows an optimal representation of CSPs, however it does not capture the property of arc consistency which is the key of almost all the enumerative CSP algorithms. We introduce in the following a simple inference rule which applies on the *CGNF* encoding C of a CSP P and prove that arc consistency is achieved with a linear complexity by application of the rule on C until saturation.

4 A New Inference Rule for the CGNF Encoding

The rule is based on the preserved CSP structure represented by both the domain and the constraint clauses.

4.1 Definition of the Inference Rule IR

Let P be a CSP, C its *CGNF* encoding, C_{C_i} a generalized constraint clause, C_{D_X} a domain clause, L_C the set of literals appearing in C_{C_i} and L_D the set of literals of C_{D_X} . If $L_C \cap L_D \neq \emptyset$ then we infer each negation $\neg X_v$ of a positive literal⁴ appearing in L_D which does not appear in L_C . We have the following rule:

IR: if $L_C \cap L_D \neq \emptyset$, $X_v \in L_D$ and $X_v \notin L_C$ then $C_{D_X} \wedge C_{C_i} \vdash \neg X_v$ ⁵.

Example 1. Let $C_{D_B} = (\pm 1, B_{v_0} B_{v_1})$ be a domain clause corresponding to the CSP variable B and $C_{C_i} = (A_{v_1} \wedge B_{v_0} \wedge C_{v_0}) \vee (A_{v_0} \wedge B_{v_0} \wedge C_{v_1})$ a constraint clause corresponding to the CSP constraint C_i involving the variables $\{A, B, C\}$. The application of the rule *IR* on both clauses infers $\neg B_{v_1}$. ($C_{D_B} \wedge C_{C_i} \vdash \neg B_{v_1}$)

Proposition 1. *The rule IR is sound (correct).*

Proof. Let X_v be a literal appearing in L_D but not in L_C and I a model of $C_{C_i} \wedge C_{D_X}$. C_{C_i} is a disjunction of conjunctions f_i and each conjunction f_i contains one literal of L_D . At least one of the conjunctions f_i , say f_j is satisfied by I since I is a model of C_{C_i} . Thus, there is a literal $X_{v'}$ ($X_{v'} \neq X_v$ since $X_v \notin L_C$) of f_j appearing in L_D , such that $I[X_{v'}] = \text{true}$. Because of the mutual exclusion of of literal of C_{D_X} , the $X_{v'}$ is the single literal of L_D satisfied by I . Thus, $I[\neg X_v] = \text{true}$ and I is a model of $\neg X_v$.

4.2 The Inference Rule and Arc-Consistency

A CSP P is arc consistent iff all its domains are arc consistent. A domain $D_{X_{i_1}}$ is arc consistent iff for each value v_{i_1} of $D_{X_{i_1}}$ and for each k-arity constraint C_j involving the variables $\{X_{i_1}, \dots, X_{i_k}\}$, there exists a tuple $(v_{i_2}, \dots, v_{i_k}) \in D_{X_{i_2}} \times \dots \times D_{X_{i_k}}$ such that $(v_{i_1}, v_{i_2}, \dots, v_{i_k}) \in R_j$. We use the inference rule *IR* on the *CGNF* encoding C of a CSP P to achieve arc-consistency. We show that by applying *IR* on C until saturation (a fixed point is reached) and by propagating each inferred negative literal we maintain arc-consistency on C . Since, this operation is based on unit propagation, it can be done in a linear time complexity.

⁴ The *CGNF* encoding of a CSP contains only positive literals.

⁵ \vdash denotes logical inference.

Proposition 2. *Let P be a CSP and C its CGNF encoding. A value $v \in D_Y$ is removed by enforcing arc-consistency on P iff the negation $\neg Y_v$ of the corresponding Boolean variable is inferred by application of IR to C .*

Proof. Let v be a value of the domain D_Y which does not verify arc-consistency. There is at least one constraint C_j involving Y such that v does not appear in any of the allowed tuples of the corresponding relation R_j . The Boolean variable Y_v does not appear in the associated constraint clause C_{C_j} , but it appears in the domain clause C_{D_Y} associated to D_Y . By applying the rule IR on both C_{D_Y} and C_{C_j} we infer $\neg Y_v$. The proof of the converse can be done in the same way.

Theorem 2. *Let P be a CSP and C its CGNF encoding. The saturation of IR on C and the propagation of all inferred literals is equivalent to enforcing arc-consistency on the original CSP P .*

Proof. Is a consequence of proposition 2.

4.3 Arc-Consistency by Application of IR

To perform arc consistency on C by applying IR , we need to define some data structures to implement the inference rule. We suppose that the nd Boolean variables of C are encoded by the first integers $[1..nd]$. We define a table OCC_j of size ad for each constraint clause C_{C_j} such that $OCC_j[i]$ gives the number of occurrences of the variable i in C_{C_j} . There is a total of m such tables corresponding to the m constraint clauses. If $OCC_j[i] = 0$ for some $i \in \{1..nd\}$ and some $j \in \{1..m\}$ then the negation $\neg i$ of the Boolean variable i is inferred by IR . This data structure adds to the space complexity a factor of $O(mad)$. The total space complexity of C is $O(mad^a + nd + mad)$, but the factors nd and mad are always lower than the mad^a factor, and the space complexity of C remains $O(mad^a)$.

The principle of the arc-consistency method consists first in reading the m tables OCC_j to detect the variables $i \in [1..nd]$ having a number of occurrences equal to zero ($OCC_j[i] = 0$). This is achieved by the steps 3 to 9 of algorithm 1 in $O(mad)$, since there are m tables of size ad . After that we apply unit propagation on the detected variables, and propagate the effect until saturation (i.e no new variable i having $OCC_j[i] = 0$ is detected). The procedure of arc-consistency is sketched in Algorithm 1. This procedure calls the procedure *Propagate* described in algorithm 2.

The complexity of the arc consistency procedure is mainly given by the propagation of the effect of literals of the list L (lines 10 to 13 of algorithm 1). It is easy to see that in the worst case there will be nd calls to the procedure *Propagate*. All the propagations due to the previous calls are performed in $O(mad^a)$ in the worst case. Indeed, there are at most d^a conjunctions of a literals for each constraint clause of C . The total number of conjunctions treated in line 4 of algorithm 2 can not exceed md^a since each considered conjunction f in line 3 is suppressed in line 4 (f is interpreted to false). As there is a literals by conjunction f , the total propagation is done in $O(mad^a)$. Thus, the complexity of the arc consistency procedure is $O(mad^a + mad)$. But the factor (mad) is necessarily smaller than mad^a and the total complexity is reduced to $O(mad^a)$. It is linear *w.r.t* the size of C .

Algorithm 1. Arc Consistency

Procedure Arc_consistency**Require:** A CGNF instance C

```

1: var  $L$  : list of literals
2:  $L = \emptyset$ 
3: for each constraint clauses  $C_{C_j}$  do
4:   for each literal  $i$  of  $OCC_j$  do
5:     if  $OCC_j[i] = 0$  then
6:       add  $i$  in  $L$ 
7:     end if
8:   end for
9: end for
10: while  $L \neq \emptyset$  and  $\square \notin C$  do
11:   extract  $l$  from  $L$ 
12:   propagate ( $C, l, L$ )
13: end while

```

Algorithm 2. Propagate

Procedure Arc_consistency**Require:** A CGNF instance C , literal i , List L

```

1: if  $i$  is not yet assigned then
2:   assign  $i$  the value false
3:   for each unassigned conjunction  $f$  of  $C$  containing  $i$  do
4:     assign  $f$  the value false
5:     for each literal  $j$  of  $f$  such that  $i \neq j$  do
6:       withdraw 1 to  $OCC_k[j]$  { $k$  is the label of the constraint clause containing  $f$ }
7:       if  $OCC_k[j] = 0$  then
8:         add  $j$  to  $L$ 
9:       end if
10:    end for
11:   end for
12: end if

```

5 Two Enumerative Methods for the CGNF Encoding

We study in the following two enumerative methods: the first one (MAC) maintains full arc consistency during the search while the second (FC) maintains partial arc consistency as does the classical Forward Checking method in CSPs [6]. Both methods perform Boolean enumeration and simplification. They are based on an adaptation of the DP procedure to the *CGNF* encoding.

5.1 The MAC Method

MAC starts by a first call to the *Arc-consistency* algorithm (Figure 1) to verify arc consistency at the root of the search tree. It then calls the procedure *Propagate* described in Figure 2 at each node of the search tree to maintain arc consistency during the search.

Algorithm 3. MAC

Procedure MAC

Require: A CGNF instance C

```

1: Arc consistency( $C$ )
2: if  $\square \in C$  then
3:   return unsatisfiable
4: else
5:   choose a literal  $l \in C$ 
6:   if Satisfiable( $C, l, true$ ) then
7:     return satisfiable
8:   else if Satisfiable( $C, l, false$ ) then
9:     return satisfiable
10:  else
11:    return unsatisfiable
12:  end if
13: end if

```

That is, the mono-literals of each node are inferred and their effects are propagated. The code of *MAC* is sketched in Figure 3.

5.2 The FC Method

It is easy to obtain from *MAC* a *Forward Checking (FC)* algorithm version for the CGNF encoding. The principle is the same as in *MAC* except that instead of enforcing full arc-consistency at each node of the search tree, (FC) does it only on the near neighbors of the current variable under assignment. This is done by restricting the propagation effect of the current literal assignment to only the literals of the clauses in which it appears. It is important to see that finding the *FC* version in our encoding is trivial, whereas it is not the case for n -ary CSP where six versions of *FC* are provided [3].

5.3 Heuristics for Literal Choice

Because the CGNF encodings keeps the CSP structure, We can find easily equivalent heuristics to all the well known CSP variable/value heuristics. For instance, the minimal domain heuristic (MD) which consists in choosing during the search the CSP variable whose domain is the smallest is equivalent to select in the CGNF encoding, a literal appearing in the shortest domain clause. This heuristic is implemented in both *MAC* and *FC* methods.

6 Experimentations

We experiment both *MAC* and *FC* methods and compare their performances on Ramsey problem, and on 3/4-ary randomly generated CSPs encoded in CGNF. The programs are written in *C*, compiled and run on a Windows operating system with a Pentium IV 2.8GHz processor and 1GB of RAM.

Algorithm 4. Satisfiable

Function Satisfiable**Require:** A CGNF instance C , variable l , Boolean val **Output:** Boolean {TRUE or FALSE}

```

1: var  $L$  : list of literals
2:  $L = \emptyset$ 
3: if  $val = true$  then
4:   assign  $l$  the value true
5:   Add each literal  $i \neq l$  of the domain clause containing  $l$  in  $L$ 
6:   while  $L \neq \emptyset$  and  $\square \notin C$  do
7:     extract  $i$  from  $L$ 
8:     propagate ( $C, i, L$ )
9:   end while
10: else
11:   repeat
12:      $Propagate(C, l, L)$ 
13:   until  $L \neq \emptyset$  and  $\square \notin C$ 
14: end if
15: if  $C = \emptyset$  then
16:   return TRUE
17: else if  $\square \in C$  then
18:   return FALSE
19: else
20:   Choose a literal  $p$  of  $C$ 
21:   if Satisfiable( $C, p, true$ ) then
22:     return TRUE
23:   else if Satisfiable( $C, p, false$ ) then
24:     return TRUE
25:   else
26:     return FALSE
27:   end if
28: end if

```

6.1 Ramsey Problem

Ramsey problem (see CSPlib) consists in coloring the edges of a complete graph having n vertices with k colors, such that no monochromatic triangle appears. Tables 1 shows the results of *MAC* and *FC* on some instances of Ramsey problem where $k = 3$ and n varies from 5 to 14. The first column defines the problem: Rn_k denotes a Ramsey instance with n vertices and k colors, the second and third columns show the number of nodes and the performances in CPU times of *FC* respectively *MAC* augmented by the heuristic (MD) described previously.

We can see that *FC* is better in time than *MAC* for small size instances ($n \leq 6$) but visits more nodes. However, *MAC* is better than *FC* in time and number of visited nodes when the problem size increases ($n \geq 7$). *MAC* outperforms *FC* in most of the cases for this problem.

Problem	FC + MD		MAC + MD	
	Nodes	Times	Nodes	Times
R5_3	12	0 s 48 μ s	12	0 s 85 μ s
R6_3	27	0 s 231 μ s	21	0 s 297 μ s
R11_3	237	0 s 5039 μ s	103	0 s 4422 μ s
R12_3	538	0 s 21558 μ s	130	0 s 11564 μ s
R13_3	1210	0 s 28623 μ s	164	0 s 9698 μ s
R14_3	216491	9 s 872612 μ s	6735	1 s 752239 μ s

Fig. 1. Results of *MAC* and *FC* on the Ramsey problem

6.2 Random Problems

The second class of problems is randomly generated CSPs. We carried experiment on 3-ary random CSPs where both the number of CSP variables and the number of values is 10. Our generator is an adaptation to the CGNF encoding of the Bessière et al CSP generator. It uses five parameters: n the number of variables, d the size of the domains, a the constraint arity, $dens$ the constraint density which is the ratio of the number of constraint to the maximum number of possible constraints, t the constraint tightness which is the proportion of the forbidden tuples of each constraints. The random output CSP instances are given in the CGNF encoding.

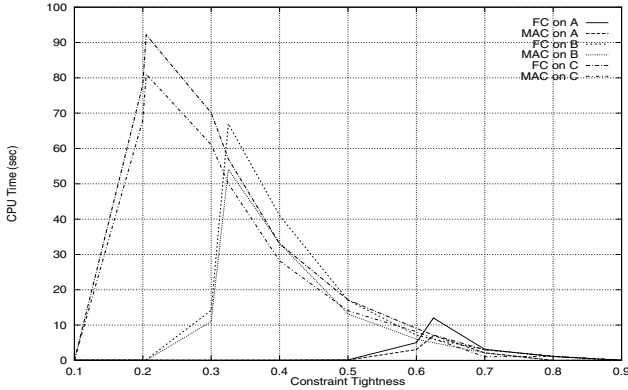


Fig. 2. Results of *MAC* and *FC* on 3-ary CSPs having the densities: A: dens=0,20, B: dens=0,50 and C: dens=0,83

Figure 2 shows the average curves representing the CPU time of *MAC* and *FC*, both augmented by the *MD* heuristic, with respect to a variation of the tightness on 3-ary CSPs. The two curves on the right are those of *MAC* and *FC* corresponding to weak densities (the class A: dens=0,20), the two ones in the middle are those corresponding to average densities (the class B: dens=0,50), and the two ones on the left are those corresponding to high densities (the class C: dens=0,83). We can see that, the peak of difficulty of the hard region of each problem class matches with a critical value of the tightness, and the hardness increases as the density increases. The smaller the density is, the

greater the corresponding critical tightness is. We can also see that *MAC* beats definitely *FC* on the three classes of problems. These results seem to compare with those obtained by the generalized *FC* [3] on 3-ary CSPs having equivalent parameters (n , d , $dens$, t).

7 Related Works and Discussion

Our approach is different from the previous works [8,7,2], since it is not a translation from CSP to SAT. It consists in a generalization of the CNF representation which allows a natural and optimal encoding for n-ary CSPs keeping the CSP structure. The purpose of this paper is first to provide a more general framework including both SAT and CSP and which captures their advantages, then introduce new promising methods to solve problems expressed in this framework. We are not interested for the moment in code optimization and heuristics to compete other optimized methods. We just provide a first implementation, whose results look to compare well with those of the nFCs for n-ary CSPs given in [3]. But, it seems that most of the background (like *non-chronological back-tracking*, or *clause recording*) added to DP procedure to produce sophisticated SAT solvers like *Chaff*[9] can be adapted for our methods. Such optimizations would increase the performance of our methods, this question will be investigated in a future work.

8 Conclusion

We studied a generalization of the known *CNF* representation which allows a compact Boolean encoding for n-ary CSPs. We showed that the size of a CSP in this encoding is identical to the one of its original representation. We implemented a new inference rule whose application until saturation achieves arc-consistency for n-ary CSPs expressed in the Boolean encoding with a linear time complexity. Two enumerative methods are proposed: the first one (*MAC*) maintains full arc-consistency on each node of the search tree while the second (*FC*) performs partial arc-consistency. Both methods are well known in CSPs and are found easily in our Boolean encoding. These methods are experimented on some instances of Ramsey problem and randomly generated 3-ary CSPs and the obtained results showed that maintaining full arc-consistency in the Boolean encoding is the best idea. These results are promising, but code optimizations are necessary to compete with sophisticated SAT solvers. As a future work, we are looking to extend the inference rule to achieve path consistency in the Boolean encoding. An other interesting point is to look for polynomial restrictions of the Boolean encoding. On the other hand, detecting and breaking symmetries in the Boolean encoding may increase the performances of the defined enumerative methods.

References

1. H. Bennaceur. The satisfiability problem regarded as constraint satisfaction problem. *Proceedings of the European Conference on Artificial Intelligence (ECAI'96)*, pages 155–159, 1996.
2. C. Bessière, E. Hebrard, and T. Walsh. Local consistency in sat. *International Conference on Theory and Application of satisfiability testing (SAT'03)*, pages 400–407, 2003.

3. C. Bessière, P. Meseguer, E. C. Freuder, and J. Larrosa. On forward checking for non-binary constraint satisfaction. *Journal of Artificial Intelligence*, 141:205–224, 2002.
4. M. Davis and H. Putnam. A computing procedure for quantification theory. *Journal of ACM* 7, pages 201–215, 1960.
5. E. Goldberg and Y. Novikov. Berkmin: A fast and robust sat solver. *Proceedings of the 2002 Design Automation and Test in Europe*, pages 142–149, 2002.
6. R.M. Haralick and G.L. Elliott. Increasing tree search efficiency for constraint satisfaction problems. *Journal of Artificial Intelligence*, 14:263–313, 1980.
7. S. Kasif. On the parallel complexity of discrete relaxation in constraint satisfaction networks. *Journal of Artificial Intelligence*, 45:275–286, 1990.
8. J. De Kleer. A comparison of atms and csp techniques. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'89)*, pages 290–296, 1989.
9. S. Malik, Y. Zhao, C. F. Madigan, L. Zhang, and M. W. Moskewicz. Chaff: Engineering an efficient sat solver. *Proceedings of the 38th conference on Design automation (IEEE 2001)*, pages 530–535, 2001.
10. U. Montanari. Networks of constraints : Fundamental properties and application to picture processing. *Journal Inform. Sci.*, 9(2):95–132, 1974.
11. D. Sabin and E. Freuder. Understanding and improving the mac algorithm. *Proceedings of International Conference on Principles and Practice of Constraint Programming (CP'97)*, pages 167–181, 1997.
12. J.M. Silva and K.A. Sakallah. Grasp – a new search algorithm for satisfiability. *Proceedings of International Conference on Computer-Aided Design (IEEE 1996)*, pages 220–227, 1996.
13. T. Walsh. Sat v csp. *Proceedings of International Conference on Principles and Practice of Constraint Programming (CP'00)*, pages 441–456, 2000.

A Constructive Hybrid Algorithm for Crew Pairing Optimization

Broderick Crawford^{1,2}, Carlos Castro², and Eric Monfroy^{2,3,*}

¹ Pontificia Universidad Católica de Valparaíso, Chile

`FirstName.Name@ucv.cl`

² Universidad Técnica Federico Santa María, Valparaíso, Chile

`FirstName.Name@inf.utfsm.cl`

³ LINA, Université de Nantes, France

`FirstName.Name@univ-nantes.fr`

Abstract. In this paper, we focus on the resolution of Crew Pairing Optimization problem that is very visible and economically significant. Its objective is to find the best schedule, i.e., a collection of crew rotations such that each airline flight is covered by exactly one rotation and the costs are reduced to the minimum. We try to solve it with Ant Colony Optimization algorithms and Hybridizations of Ant Colony Optimization with Constraint Programming techniques. We give an illustrative example about the difficulty of pure Ant Algorithms solving strongly constrained problems. Therefore, we explore the addition of Constraint Programming mechanisms in the construction phase of the ants, so they can complete their solutions. Computational results solving some test instances of Airline Flight Crew Scheduling taken from NorthWest Airlines database are presented showing the advantages of using this kind of hybridization.

Keywords: Ant Colony Optimization, Constraint Programming, Hybrid Algorithm, Crew Pairing Optimization, Set Partitioning Problem.

1 Introduction

The Crew Pairing Optimization has been investigated for many years and this problem continues challenging both scientists and software engineers. The basic problem is to partition a given schedule of airline flights into individual flight sequences called pairings. A pairing is a sequence of flight legs for an unspecified crew member starting and finishing at the same city.

The pairing problem can be formulated as a Set Partitioning Problem (SPP) or equality-constrained as a Set Covering Problem (SCP), in this formulation the rows are flights and the columns are pairings. The optimization problem is to select the partition of minimum cost from a pool of candidate pairings. SPP and

* The authors have been partially supported by the project INRIA-CONICYT VANANAA. The third author has also been partially supported by the Chilean National Science Fund through the project FONDECYT 1060373.

SCP are two types of problems that can model several real life situations [4,14]. In this work, we solve some test instances of Airline Flight Crew Scheduling with Ant Colony Optimization (ACO) algorithms and some hybridizations of ACO with Constraint Programming (CP) techniques like Forward Checking.

There exist some problems for which the effectiveness of ACO is limited, among them the strongly constrained problems. Those are problems for which neighbourhoods contain few solutions, or none at all, and local search has a very limited use. Probably, the most significant of those problems is the SPP and a direct implementation of the basic ACO framework is unable of obtaining feasible solutions for many SPP standard tested instances [21]. The best performing metaheuristic for SPP is a genetic algorithm due to Chu and Beasley [9,6]. There already exists some first approaches applying ACO to the SCP. In [1,19] ACO has only been used as a construction algorithm and the approach has only been tested on some small SCP instances. More recent works [18,20,17] apply Ant Systems to the SCP and related problems using techniques to remove redundant columns and local search to improve solutions. Taking into account these results, it seems that the incomplete approach of Ant Systems could be considered as a good alternative to solve these problems when complete techniques are not able to get the optimal solution in a reasonable time.

In this paper, we explore the addition of a lookahead mechanism to the two main ACO algorithms: Ant System (AS) and Ant Colony System (ACS). Trying to solve larger instances of SPP with AS or ACS implementations derives in a lot of unfeasible labelling of variables, and the ants can not obtain complete solutions using the classic transition rule when they move in their neighbourhood. In this paper, we propose the addition of a lookahead mechanism in the construction phase of ACO thus only feasible partial solutions are generated. The lookahead mechanism allows the incorporation of information about the instantiation of variables after the current decision. This idea differs from the one proposed by [23] and [16], those authors propose a lookahead function evaluating the pheromone in the Shortest Common Supersequence Problem and estimating the quality of a partial solution of a Industrial Scheduling Problem, respectively. So, there are two main motivations for our work. On one hand, we try to improve the performance of ACO algorithms when dealing with hard instances of SPP through the incorporation of local analysis of the constraints. On the other hand, we are interested in the development of robust algorithms integrating complete as well as incomplete techniques, because we are convinced that this is a very promising approach to deal with hard combinatorial problems.

This paper is organised as follows: Section 2 is dedicated to the presentation of the problem and its mathematical model. In Section 3, we describe the applicability of the ACO algorithms for solving SPP and an example of Constraint Propagation is given. In Section 4, we present the basic concepts to adding Constraint Programming techniques to the two basic ACO algorithms: AS and ACS. In Section 5, we present results when adding Constraint Programming techniques to the two basic ACO algorithms to solve some Airline Flight Crew Scheduling taken from NorthWest Airlines benchmarks available in the

OR-Library of Beasley [5]. Finally, in Section 6 we conclude the paper and give some perspectives for future research.

2 Problem Description

The resource planning in airlines is a very complex task, and without considering the fuel costs, the most important direct operating cost is the personnel. The planning and scheduling of crews is usually considered as two optimization problems: the crew pairing problem and the crew assignment problem (or rostering problem). In this paper we focus on the pairing problem. The crew costs depend on the quality of the solution to the pairing problem as well as the assignment, but since it is not possible to make up for poor pairings in the assignment problem, it is reasonable to expect that savings in the pairings problem will lead to savings in total crew costs [2].

The main difficulty in modelling the crew pairing problem is that the set of feasible pairings is very difficult to characterize mathematically in any other way than by enumeration. In addition, the cost of a pairing is usually a complex function of its components. Therefore, all published methods attempt to separate the problem of generating pairings from the problem of selecting the best subset of these pairings. The remaining optimization problem is then modelled under the assumption that the set of feasible pairings and their costs are explicitly available, and can be expressed as a Set Partitioning Problem. The SPP model is valid for the daily problem as well as the weekly problem and the fully dated problem.

SPP is the NP-complete problem of partitioning a given set into mutually independent subsets while minimizing a cost function defined as the sum of the costs associated to each of the eligible subsets. In the SPP matrix formulation we are given a $m \times n$ matrix $A = (a_{ij})$ in which all the matrix elements are either zero or one. Additionally, each column is given a non-negative cost c_j . We say that a column j can cover a row i if $a_{ij} = 1$. Let J denotes the set of the columns and x_j a binary variable which is one if column j is chosen and zero otherwise. The SPP can be defined formally as follows:

$$\text{Minimize} \quad f(x) = \sum_{j=1}^n c_j \times x_j \quad (1)$$

$$\text{Subject to} \quad \sum_{j=1}^n a_{ij} \times x_j = 1; \quad \forall i = 1, \dots, m \quad (2)$$

These constraints enforce that each row is covered by exactly one column. The SPP has been studied extensively over the years because of its many real world applications. One of the well known applications is airline crew pairing. In this formulation, each row represents a flight leg that must be scheduled. The columns represent pairings. Each pairing is a sequence of flights to be covered by a single crew over a 2 to 3 day period. It must begin and end in the base city where the crew resides [24].

3 Ant Colony Optimization for Set Partitioning Problems

In this section, we briefly present ACO algorithms and give a description of their use to solve SPP. More details about ACO algorithms can be found in [11,12].

The basic idea of ACO algorithms comes from the capability of real ants to find shortest paths between the nest and food source. From a Combinatorial Optimization point of view, the ants are looking for *good solutions*. Real ants cooperate in their search for food by depositing pheromone on the ground. An artificial ant colony simulates this behavior implementing artificial ants as parallel processes whose role is to build solutions using a randomized constructive search driven by pheromone trails and heuristic information of the problem. An important topic in ACO is the adaptation of the pheromone trails during algorithm execution to take into account the cumulated search experience: reinforcing the pheromone associated with good solutions and considering the *evaporation* of the pheromone on the components over time in order to avoid premature convergence. ACO can be applied in a very straightforward way to SPP. The columns are chosen as the solution components and have associated a cost and a pheromone trail [13]. Each column can be visited by an ant only once and then a final solution has to cover all rows. A walk of an ant over the graph representation corresponds to the iterative addition of columns to the partial solution obtained so far. Each ant starts with an empty solution and adds columns until a cover is completed. A pheromone trail τ_j and a heuristic information η_j are associated to each eligible column j . A column to be added is chosen with a probability that depends of pheromone trail and the heuristic information. The most common form of the ACO decision policy (*Transition Rule Probability*) when ants work with components is:

$$p_j^k(t) = \frac{\tau_j * \eta_j^\beta}{\sum_{l \notin S^k} \tau_l [\eta_l]^\beta} \quad \text{if } j \notin S^k \quad (3)$$

where S^k is the partial solution of the ant k . The β parameter controls how important is η in the probabilistic decision [13,20].

Pheromone trail τ_j . One of the most crucial design decisions to be made in ACO algorithms is the modelling of the set of pheromones. In the original ACO implementation for TSP the choice was to put a pheromone value on every link between a pair of cities, but for other combinatorial problems often can be assigned pheromone values to the decision variables (first order pheromone values) [13]. In this work the pheromone trail is put on the problems component (each eligible column j) instead of the problems connections. And setting a good pheromone quantity is not a trivial task either. The quantity of pheromone trail laid on columns is based on the idea: *the more pheromone trail on a particular item, the more profitable that item is* [19]. Then, the pheromone deposited in each component will be in relation to its frequency in the ants solutions. In this work we divided this frequency by the number of ants obtaining better results.

Heuristic information η_j . In this paper we use a dynamic heuristic information that depends on the partial solution of an ant. It can be defined as $\eta_j = \frac{e_j}{c_j}$, where e_j is the so called cover value, that is, the number of additional rows covered when adding column j to the current partial solution, and c_j is the cost of column j . In other words, the heuristic information measures the unit cost of covering one additional row. An ant ends the solution construction when all rows are covered. Figure 1 describes the basic ACO algorithm to solve SPP.

```

1  Procedure ACO_for_SPP
2  Begin
3    InitParameters();
4    While (remain iterations) do
5      For k := 1 to nants do
6        While (solution is not completed)
7          Choose next Column j with Transition Rule Probability
8          AddColumnToSolution(j)
9          AddColumnToTabuList(j)
10       EndWhile
11     EndFor
12     UpdateOptimum();
13     UpdatePheromone();
14   EndWhile
15   Return best_solution_founded
16 End.
```

Fig. 1. ACO algorithm for SPP

In this work, we use two instances of ACO: Ant System (AS) and Ant Colony System (ACS) algorithms, the original and the most famous algorithms in the ACO family [13]. ACS improves the search of AS using: a different transition rule in the constructive phase, exploiting the heuristic information in a more rude form, using a list of candidates to future labelling and using a different treatment of pheromone. ACS has demonstrated better performance than AS in a wide range of problems [12].

Trying to solve larger instances of SPP with the original AS or ACS implementation derives in a lot of unfeasible labelling of variables, and the ants can not obtain complete solutions. In this paper we explore the addition of a lookahead mechanism in the construction phase of ACO thus only feasible solutions are generated.

A direct implementation of the basic ACO framework is incapable of obtaining feasible solution for many SPP instances. An example will be given in order to explain the ACO difficulties solving SPP. In [24] is showed Table 1 with a Flight Schedule for American Airlines. The table enumerates possible pairings, or sequence of flights to be covered by a single crew over a 2 to 3 day period, and its costs. A pairing must begin and end in the base city where the crew resides. For example, pairing $j = 1$ begins at a known city (Miami in the [24] example) with flight 101 (Miami-Chicago). After a layover in Chicago the crew covers flight 203 (Chicago-Dallas) and then flight 406 (Dallas-Charlotte) to Charlotte. Finally, flight 308 (Charlotte-Miami) returns them to Miami. The total cost of pairing $j = 1$ is \$ 2900.

Table 1. Possible Pairings for AA Example

Pairing j	Flight Sequence	Cost \$
1	101-203-406-308	2900
2	101-203-407	2700
3	101-204-305-407	2600
4	101-204-308	3000
5	203-406-310	2600
6	203-407-109	3150
7	204-305-407-109	2550
8	204-308-109	2500
9	305-407-109-212	2600
10	308-109-212	2050
11	402-204-305	2400
12	402-204-310-211	3600
13	406-308-109-211	2550
14	406-310-211	2650
15	407-109-211	2350

Having enumerated a list of pairings like Table 1, the remaining task is to find a minimum total cost collection of columns staffing each flight exactly once. Defining the decision variables x_j equal to 1 if pairing j is chosen and 0 otherwise, the corresponding SPP model must to be solved.

$$\text{Minimize } 2900x_1 + 2700x_2 + 2600x_3 + 3000x_4 + 2600x_5 + 3150x_6 + 2550x_7 + 2500x_8 + 2600x_9 + 2050x_{10} + 2400x_{11} + 3600x_{12} + 2550x_{13} + 2650x_{14} + 2350x_{15}$$

Subject to

$$\begin{aligned}
x_1 + x_2 + x_3 + x_4 &= 1 & (\text{flight } 101) \\
x_6 + x_7 + x_8 + x_9 + x_{10} + x_{13} + x_{15} &= 1 & (\text{flight } 109) \\
x_1 + x_2 + x_5 + x_6 &= 1 & (\text{flight } 203) \\
x_3 + x_4 + x_7 + x_8 + x_{11} + x_{12} &= 1 & (\text{flight } 204) \\
x_{12} + x_{13} + x_{14} + x_{15} &= 1 & (\text{flight } 211) \\
x_9 + x_{10} &= 1 & (\text{flight } 212) \\
x_3 + x_7 + x_9 + x_{11} &= 1 & (\text{flight } 305) \\
x_1 + x_4 + x_8 + x_{10} + x_{13} &= 1 & (\text{flight } 308) \\
x_5 + x_{12} + x_{14} &= 1 & (\text{flight } 310) \\
x_{11} + x_{12} &= 1 & (\text{flight } 402) \\
x_1 + x_5 + x_{13} + x_{14} &= 1 & (\text{flight } 406) \\
x_2 + x_3 + x_6 + x_7 + x_9 + x_{15} &= 1 & (\text{flight } 407) \\
x_j &= 0 \text{ or } 1; \quad \forall j = 1, \dots, 15
\end{aligned}$$

An optimal solution of this problem, at cost of \$ 9100, is $x_1^* = x_9^* = x_{12}^* = 1$ and all other $x_j^* = 0$.

Applying ACO to the American Airlines Example. Each ant starts with an empty solution and adds columns until a cover is completed. But to determine

if a column actually belongs or not to the partial solution ($j \notin S^k$) is not good enough. The traditional ACO decision policy, Equation 3, does not work for SPP because the ants, in this traditional selection process of the next columns, ignore the information of the problem constraints.

For example, let us suppose that at the beginning an ant chooses the pairing or column number 14, then x_{14} is instantiated with the value 1. For instance, if x_{14} is instantiated, the consideration of the constraints that contain x_{14} may have important consequences:

- Checking constraint of flight 211, if $x_{14} = 1$ then $x_{12} = x_{13} = x_{15} = 0$.
- Checking constraint of flight 310, if $x_{14} = 1$ then $x_5 = x_{12} = 0$.
- Checking constraint of flight 406, if $x_{14} = 1$ then $x_1 = x_5 = x_{13} = 0$.
- If $x_{12} = 0$, considering the flight 402 constraint then $x_{11} = 1$.
- If $x_{11} = 1$, considering the flight 204 constraint then $x_3 = x_4 = x_7 = x_8 = 0$; and by the flight 305 constraint then $x_3 = x_7 = x_9 = 0$.
- If $x_9 = 0$, considering the flight 212 constraint then $x_{10} = 1$.
- If $x_{10} = 1$, by the flight 109 constraint $x_6 = x_7 = x_8 = x_9 = x_{13} = x_{15} = 0$; and considering the flight 308 constraint $x_1 = x_4 = x_8 = x_{13} = 0$.

All the information above, where the only variable uninstantiated after a simple propagation of constraints was x_2 , is ignored by the probabilistic transition rule of the ants. And in the worst case, in the iterative steps is possible to assign values to some variable that will make impossible to obtain complete solutions.

The procedure that we showed above is similar to the Constraint Propagation technique.

Constraint Propagation is an efficient inference mechanism based on the use of the information in the constraints that can be found under different names: Constraint Relaxation, Filtering Algorithms, Narrowing Algorithms, Constraint Inference, Simplification Algorithms, Label Inference, Local Consistency Enforcing, Rules Iteration, Chaotic Iteration. Constraint Propagation embeds any reasoning which consists in explicitly forbidding values or combinations of values for some variables of a problem because a given subset of its constraints cannot be satisfied otherwise [7].

The algorithm proceeds as follows: when a value is assigned to a variable, the algorithm recomputes the possible value sets and assigned values of all its dependent variables (variable that belongs to the same constraint). This process continues recursively until no more changes can be done. More specifically, when a variable x_m changes its value, the algorithm evaluates the domain expression of each variable x_n dependent on x_m . This may generate a new set of possible values for x_n . If this set changes, a constraint is evaluated selecting one of the possible values as the new assigned value for x_n . It causes the algorithm to recompute the values for further downstream variables. In the case of binary variables the constraint propagation works very fast in strongly constrained problems like SPP.

The two basic techniques of Constraint Programming are Constraint Propagation and Constraint Distribution. The problem cannot be solved using Constraint

Propagation alone, Constraint Distribution or Search is required to reduce the search space until Constraint Propagation is able to determine the solution. Constraint Distribution splits a problem into complementary cases once Constraint Propagation cannot advance further. By iterating propagation and distribution, propagation will eventually determine the solutions of a problem [3].

4 ACO with Constraint Programming

Recently, some efforts have been done in order to integrate Constraint Programming techniques to ACO algorithms [22,8,15]. An hybridization of ACO and CP can be approached from two directions: we can either take ACO or CP as the base algorithm and try to embed the respective other method into it. A form to integrate CP into ACO is to let it reduce the possible candidates among the not yet instantiated variables participating in the same constraints that the actual variable. A different approach would be to embed ACO within CP. The point at which ACO can interact with CP is during the labelling phase, using ACO to learn a value ordering that is more likely to produce good solutions.

In this work, ACO use CP in the variable selection (when adding columns to partial solution). The CP algorithm used in this paper is Forward Checking with Backtracking. The algorithm is a combination of Arc Consistency Technique and Chronological Backtracking [10]. It performs Arc Consistency between pairs of a not yet instantiated variable and an instantiated variable, i.e., when a value is assigned to the current variable, any value in the domain of a future variable which conflicts with this assignment is removed from the domain.

Adding Forward Checking to ACO for SPP means that columns are chosen if they do not produce any conflict with the next column to be chosen. In other words, the Forward Checking search procedure guarantees that at each step of the search, all the constraints between already assigned variables and not yet assigned variables are arc consistency.

This reduces the search tree and the overall amount of computational work done. But it should be noted that in comparison with pure ACO algorithm, Forward Checking does additional work when each assignment is intended to be added to the current partial solution. Arc consistency enforcing always increases the information available on each variable labelling. Figure 2 describes the hybrid ACO+CP algorithm to solve SPP.

5 Experiments and Results

Table 2 presents the results when adding Forward Checking to the basic ACO algorithms for solving test instances taken from the Orlib [5]. The first five columns of Table 2 present the problem code, the number of rows (constraints), the number of columns (decision variables), the best known solution for each instance, and the density (percentage of ones in the constraint matrix) respectively. The next two columns present the cost obtained when applying AS and

```

1  Procedure ACO+CP_for_SPP
2  Begin
3    InitParameters();
4    While (remain iterations) do
5      For k := 1 to nants do
6        While (solution is not completed) and TabuList <> J do
7          Choose next Column j with Transition Rule Probability
8          For each Row i covered by j do          /* constraints with j      */
9            feasible(i):= Posting(j);              /* Constraint Propagation */
10         EndFor
11         If feasible(i) for all i then AddColumnToSolution(j)
12         else Backtracking(j); /* set j uninstantiated */
13         AddColumnToTabuList(j);
14       EndWhile
15     EndFor
16     UpdateOptimum();
17     UpdatePheromone();
18   EndWhile
19   Return best_solution_founded
20 End.

```

Fig. 2. ACO+CP algorithm for SPP**Table 2.** ACO with Forward Checking

Problem	Rows(Constraints)	Columns(Variables)	Optimum	Density	AS	ACS	AS+FC	ACS+FC
sppnw06	50	6774	7810	18.17	9200	9788	8160	8038
sppnw08	24	434	35894	22.39	X	X	35894	36682
sppnw09	40	3103	67760	16.20	70462	X	70222	69332
sppnw10	24	853	68271	21.18	X	X	X	X
sppnw12	27	626	14118	20.00	15406	16060	14466	14252
sppnw15	31	467	67743	19.55	67755	67746	67743	67743
sppnw19	40	2879	10898	21.88	11678	12350	11060	11858
sppnw23	19	711	12534	24.80	14304	14604	13932	12880
sppnw26	23	771	6796	23.77	6976	6956	6880	6880
sppnw32	19	294	14877	24.29	14877	14886	14877	14877
sppnw34	20	899	10488	28.06	13341	11289	10713	10797
sppnw39	25	677	10080	26.55	11670	10758	11322	10545
sppnw41	17	197	11307	22.10	11307	11307	11307	11307

ACS, and the last two columns present the results combining AS and ACS with Forward Checking. An entry of "X" in the table means no feasible solution was found. The algorithms have been run with the following parameters settings: influence of pheromone (α)=1.0, influence of heuristic information (β)=0.5 and evaporation rate (ρ)=0.4 as suggested in [19,20,13]. The number of ants has been set to 120 and the maximum number of iterations to 160, so that the number of generated candidate solutions is limited to 19.200. For ACS the list size was 500 and Q_0 =0.5. Algorithms were implemented using ANSI C, GCC 3.3.6, under Microsoft Windows XP Professional version 2002.

The effectiveness of Constraint Programming is showed to solve SPP, because the SPP is so strongly constrained the stochastic behaviour of ACO can be improved with lookahead techniques in the construction phase, so that almost only feasible partial solutions are induced. In the original ACO implementation the SPP solving derives in a lot of unfeasible labelling of variables, and the ants can not complete solutions.

6 Conclusions and Future Directions

We have successfully combined Constraint Programming and ACO for the problem of set partitioning solving benchmarks of data sets. Our main conclusion from this work is that we can improve ACO with CP. The concept of Arc Consistency plays an essential role in Constraint Programming as a problem simplification operation and as a tree pruning technique during search through the detection of local inconsistencies among the uninstantiated variables. We have shown that it is possible to add Arc Consistency to any ACO algorithms and the computational results confirm that the performance of ACO can be improved with this type of hybridisation. Anyway, a complexity analysis should be done in order to evaluate the cost we are adding with this kind of integration. We strongly believe that this kind of integration between complete and incomplete techniques should be studied deeply.

Future versions of the algorithm will study the pheromone treatment representation and the incorporation of available techniques in order to reduce the input problem (Pre Processing) and improve the solutions given by the ants (Post Processing). The ants solutions may contain expensive components which can be eliminated by a fine tuning heuristic after the solution, then we will explore Post Processing procedures, which consists in the identification and replacement of the columns of the ACO solution in each iteration by more effective columns. Besides, the ants solutions can be improved by other local search methods like Hill Climbing, Simulated Annealing or Tabu Search.

References

1. D. Alexandrov and Y. Kochetov. Behavior of the ant colony algorithm for the set covering problem. In *Proc. of Symp. Operations Research*, pages 255–260. Springer Verlag, 2000.
2. E. Andersson, E. Housos, N. Kohl, and D. Wedelin. Crew pairing optimization. In Y. G., editor, *Operations Research in the Airline Industry*. Kluwer Academic Publishing, 1998.
3. K. R. Apt. *Principles of Constraint Programming*. Cambridge University Press, 2003.
4. E. Balas and M. Padberg. Set partitioning: A survey. *SIAM Review*, 18:710–760, 1976.
5. J. E. Beasley. Or-library:distributing test problem by electronic mail. *Journal of Operational Research Society*, 41(11):1069–1072, 1990.
6. J. E. Beasley and P. C. Chu. A genetic algorithm for the set covering problem. *European Journal of Operational Research*, 94(2):392–404, 1996.
7. C. Bessiere. Constraint propagation. Technical Report 06020, LIRMM, March 2006. also as Chapter 3 of the Handbook of Constraint Programming, F. Rossi, P. van Beek and T. Walsh eds. Elsevier 2006.
8. C. Castro, M. Moossen, and M. C. Riff. A cooperative framework based on local search and constraint programming for solving discrete global optimisation. In *Advances in Artificial Intelligence: 17th Brazilian Symposium on Artificial Intelligence, SBIA 2004*, volume 3171 of *Lecture Notes in Artificial Intelligence*, pages 93–102, Sao Luis, Brazil, October 2004. Springer.

9. P. C. Chu and J. E. Beasley. Constraint handling in genetic algorithms: the set partitioning problem. *Journal of Heuristics*, 4:323–357, 1998.
10. R. Dechter and D. Frost. Backjump-based backtracking for constraint satisfaction problems. *Artificial Intelligence*, 136:147–188, 2002.
11. M. Dorigo, G. D. Caro, and L. M. Gambardella. Ant algorithms for discrete optimization. *Artificial Life*, 5:137–172, 1999.
12. M. Dorigo and L. M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.
13. M. Dorigo and T. Stutzle. *Ant Colony Optimization*. MIT Press, USA, 2004.
14. A. Feo, G. Mauricio, and A. Resende. A probabilistic heuristic for a computationally difficult set covering problem. *OR Letters*, 8:67–71, 1989.
15. F. Focacci, F. Laburthe, and A. Lodi. Local search and constraint programming. In *Handbook of metaheuristics*. Kluwer, 2002.
16. C. Gagne, M. Gravel, and W. Price. A look-ahead addition to the ant colony optimization metaheuristic and its application to an industrial scheduling problem. In J. S. et al., editor, *Proceedings of the fourth Metaheuristics International Conference MIC'01*, pages 79–84, July 2001.
17. X. Gandibleux, X. Delorme, and V. T'Kindt. An ant colony algorithm for the set packing problem. In M. D. et al., editor, *ANTS 2004*, volume 3172 of *LNCS*, pages 49–60. SV, 2004.
18. R. Hadji, M. Rahoual, E. Talbi, and V. Bachelet. Ant colonies for the set covering problem. In M. D. et al., editor, *ANTS 2000*, pages 63–66, 2000.
19. G. Leguizamón and Z. Michalewicz. A new version of ant system for subset problems. In *Congress on Evolutionary Computation, CEC'99*, pages 1459–1464, Piscataway, NJ, USA, 1999. IEEE Press.
20. L. Lessing, I. Dumitrescu, and T. Stutzle. A comparison between aco algorithms for the set covering problem. In M. D. et al., editor, *ANTS 2004*, volume 3172 of *LNCS*, pages 1–12. SV, 2004.
21. V. Maniezzo and M. Milandri. An ant-based framework for very strongly constrained problems. In M. D. et al., editor, *ANTS 2002*, volume 2463 of *LNCS*, pages 222–227. SV, 2002.
22. B. Meyer and A. Ernst. Integrating aco and constraint propagation. In M. D. et al., editor, *ANTS 2004*, volume 3172 of *LNCS*, pages 166–177. SV, 2004.
23. R. Michel and M. Middendorf. An island model based ant system with lookahead for the shortest supersequence problem. In *Lecture notes in Computer Science*, Springer Verlag, volume 1498, pages 692–701, 1998.
24. R. L. Rardin. *Optimization in Operations Research*. Prentice Hall, 1998.

Using Local Search for Guiding Enumeration in Constraint Solving

Eric Monfroy^{1,2}, Carlos Castro¹, and Broderick Crawford^{1,3,*}

¹ Universidad Técnica Federico Santa María, Valparaíso, Chile
FirstName.Name@inf.utfsm.cl

² LINA, Université de Nantes, France
FirstName.Name@univ-nantes.fr

³ Pontificia Universidad Católica de Valparaíso, Chile
FirstName.Name@ucv.cl

Abstract. In Constraint Programming, enumeration strategies (selection of a variable and a value of its domain) are crucial for resolution performances. We propose to use Local Search for guiding enumeration: we extend the common variable selection strategies of constraint programming and we achieve the value selection based on a Local Search. The experimental results are rather promising.

1 Introduction

Systematic backtracking enhanced with pruning of the search space has been successfully applied to combinatorial problems for decades. The main advantage of these techniques is completeness: if there is a solution they find it, and they give a proof when there is no solution. However, they do not always scale well for large problems. Incomplete methods, such as Local Search (LS) [6] explore some promising parts of the search space with respect to specific heuristics. These techniques are incomplete, but scale better to large problems.

Constraint propagation based solvers (CP) [1] are complete methods to solve Constraint Satisfaction Problems (CSP): they interleave enumeration and constraint propagation. Propagation prunes the search tree by eliminating values that cannot participate in a solution. Enumeration creates one branch by instantiating a variable ($x = v$) and another branch ($x \neq v$) for backtracking when the first branch does not contain any solution. All enumeration strategies preserving solutions are valid (e.g., first-fail, max-occurrence, brelaz, ...), but they have a huge impact on resolution efficiency. Moreover, no strategy is (one of) the best for all problems. Numerous studies have been conducted about enumeration strategies (see e.g., [3,2]).

A common idea to get more efficient and robust algorithms consists in combining resolution paradigms to take advantage of their respective assets. Hybridisations of CP and LS [5,15,14] are now more and more studied in the constraint

* The first author has been partially supported by the Chilean National Science Fund through the project FONDECYT 1060373. The authors have been partially supported by the project INRIA-CONICYT VANANAA.

programming community. Our approach is similar to [10] and [15]. However, these works address SAT problems and thus algorithms and strategies are different. [7] reports about the importance of the enumeration strategy for the heavy-tailed behaviour.

We propose a Master/Slave hybridisation in which LS guides the search by helping selections of variables and values for enumeration. Our goal is to obtain a solving process in which enumeration strategies (and more especially variable selection) impact less the solving efficiency: we want to avoid that a “bad” variable selection drastically reduces efficiency, without penalising a “good” selection strategy. Consequently, we also want to avoid heavy-tailed behaviour. The goal is thus to reduce the effect that could have a bad strategy selected by the user when this one does not know which strategy is the best adapted to his problem.

The technique consists in improving standard enumeration strategies. LS is performed before each enumeration in order 1) to improve standard variable selection strategies (e.g., first-fail) by requesting some properties variables (e.g., not conflicting variables), and 2) to provide a value for the selected variable. Each time, a LS is run on a different problem, i.e., the problem that remains after removing variables that have been instantiated in CP (either by enumeration or propagation) and instantiated constraints that are satisfied by these assignments. Obviously, it can also happen that a LS solves the remaining problem. Note that our algorithm is still complete. The experimental results we obtained with our prototype implementation are promising.

2 Background, Motivations, and Ideas

A Constraint Satisfaction Problem (CSP) is defined by a set of variables, a set of possible values for each variable (the domain of the variable), and a set of n -ary constraints. A solution is an instantiation of variables satisfying all constraints.

Constraint Propagation Based Solvers. Systematic backtracking is used in many constraint solvers. It is complete, and can be combined with constraint propagation for pruning the search space. However, it often behaves poorly for large combinatorial problems. Constraint propagation based solvers can be described by the generic algorithm of Figure 1 (similar to the one of [1]). They interleaved propagation phases with split steps. Propagation reduces the search space by removing values of variables that cannot participate to a solution of the CSP. A **split** cuts a CSP \mathcal{P} into several CSPs \mathcal{P}_i , such that the union of solutions of the \mathcal{P}_i is equal to the solutions of \mathcal{P} (preserve solutions). Each \mathcal{P}_i

```

WHILE not finished
  constraint propagation
  IF not finished THEN
    select a split and apply it
    search

```

Fig. 1. A Generic SOLVE algorithm

```

choose  $s \in S$  (an initial configuration)
 $opt \leftarrow s$  (record the best configuration)
WHILE not finished DO
    choose  $s'$  a neighbour of  $s$ 
     $s \leftarrow s'$  (move to  $s'$ )
     $opt \leftarrow s$  if  $f(s) < f(opt)$ 

```

Fig. 2. A Generic Local Search Algorithm

differs from \mathcal{P} in that the split domain is replaced by a smaller domain. The two main classes of split strategies are segmentation (split a domain into sub-domains), and enumeration (split a domain into one value and the rest of the domain). In the following, we focus on enumeration. **Search** manages the choice points (i.e., sub-CSPs created by split) by making recursive calls to the solve algorithm: it may define searches such as depth first, or breadth first; it can enforce finding one solution or all solutions; or can manage optimisation or satisfaction. **finished** is a Boolean set to true when the problem is either solved or unsatisfiable.

Local Search. While complete methods can demonstrate that a problem is not satisfiable (exhaustive exploration of the search space), incomplete methods will be ineffective in that case. However, they scale very well to large applications since they mainly rely on heuristics providing a more efficient exploration of interesting areas of the search space. This class of methods (called metaheuristics) covers a wide panel of paradigms from evolutionary algorithms to local search techniques [6] on which we focus in the following.

Local search techniques usually aim at solving optimisation problems [6]. In the context of constraint satisfaction, these methods minimize the number of violated constraints to find a solution of the CSP. A local search algorithm (see Figure 2), starting from an initial configuration, explores the search space by a sequence of moves. At each iteration, the next move corresponds to the choice of one of the so-called neighbours. This neighbourhood often corresponds to small changes of the current configuration. Moves are guided by a fitness function (f on the figure) which evaluates their benefit from the optimisation point of view, in order to reach a local optimum. The algorithm stops (Boolean finished) when a solution is found or when a maximum number of iterations is reached.

Hybridising CP and LS. A common idea to get more efficient algorithms consists in combining resolution paradigms [5,15,14]. Such combinations are more and more studied in the constraint programming community, mainly for combining CP with LS or genetic algorithms. Hybridisations have often been tackled through a Master/Slave like management, and are often related to specific problems or class of problems. Among Master/Slave approaches in which CP is the master, we can cite: local probing [9], selecting variables by LS [10], LS for improving partial assignment [12], LS for guiding the backtrack [13]. Generally, when LS is considered as the master, CP is used to improve the quality or the size of the neighbourhood [8]. Other techniques sacrifice completeness. Numerous

```

solve(CSP)
CSP'  $\leftarrow$  constraint_propagation(CSP)
IF solution(CSP') OR failed(CSP')
    THEN RETURN(CSP')
    ELSE Tentative_Val  $\leftarrow$  - LS(CSP')
        (Var,Val) $\leftarrow$  select_enumeration(CSP',VarCrit,ValCrit, Tentative_Val)
        RES  $\leftarrow$  solve(CSP'  $\wedge$  Var = Val)
        IF failed(RES)
            THEN solve(CSP'  $\wedge$  Var  $\neq$  Val)
            ELSE RETURN(RES)

```

Fig. 3. A Depth-First Left-First hybrid SOLVE algorithm for finding ONE solution

works also consider sequential or parallel hybridisation of black-box solvers [4]. Few works focus on generic hybrid solvers or hybrid frameworks [11].

The Impact of Enumeration. For the simple 10-queen problem, a good enumeration strategy directly goes to a solution performing 6 enumerations without backtracking. However, a bad strategy performs more than 800 backtracks before reaching a solution. Obviously strategies have drastically different efficiencies, often several orders of magnitude, and thus it is crucial to select a good one that unfortunately cannot be predicted in the general case.

We are interested in making good choices for enumeration, i.e., selection of a variable and a value. Our idea is to use LS to guide enumerations. There exist numerous enumeration strategies for CP, and some are known to be efficient for some problems. We thus focus on improving bad strategies without affecting good ones, and this, without knowing which are the good ones and which are the bad ones. Let us illustrate this on example. One standard strategy selects the variable with the smallest domain, and instantiates it with the first value of its domain. In our hybrid solver, we run a LS before enumeration. A possible modification of the first-fail is to select the smallest variable which is not in conflict in LS, and to assign it the tentative value given by the LS.

3 LS Guided Enumeration in CP

3.1 The Hybrid Algorithm

We designed a Master/Slave hybridisation (Figure 3) in which a LS guides enumeration in CP to quickly find a solution: a LS is performed before each enumeration to find information about variables and their tentative values. The exploration of the search tree is a depth-first left-first search. The while loop of the generic algorithm (Figure 1) does not exist anymore since we are looking for one solution. The **select_enumeration** function is the key point: not only it is based on criteria for variable and value selection, but also on the result of the LS, i.e., the tentative values of the variables. A LS finds tentative values for the non instantiated variables of the CSP considering that values assigned by previous enumeration and propagation are correct: a LS does not reconsider

previous work (propagation and instantiation); this task is left to the CP solver that will backtrack when it detects unsatisfiability of the CSP. Note that our hybrid solver remains complete.

Let us fix first some notions. A variable can only be instantiated by propagation or enumeration, but never by LS. A *tentative value* is a value given by LS to a variable which has not yet been instantiated; this value must be an element of the domain of the variable; we denote it $X \equiv v$. A *tentative CSP* is a CSP in which variables which do not have yet values are given tentative values. A *conflicting constraint* is a constraint which cannot be satisfied with the values and tentative values of the variables it contains. Consider the following CSP: $X = 3, Y \equiv 4, X > Y$. 3 is the value of X given by enumeration or propagation; 4 is the tentative value given to Y by the last LS; the constraint $X > Y$ is a conflicting constraint. A *possibly conflicting variable* (conflicting variable in short) is a not yet instantiated variable that appears in a conflicting constraint. Note that the tentative value of this variable may participate in a solution. Consider the CSP $X \in [1..5], Y \in [1..5], X > Y$ and the tentative values $X \equiv 3, Y \equiv 4$. Y is a conflicting variable since it appears in $X > Y$ which is a conflicting constraint. However, a solution of the CSP is $X = 5, Y = 4$.

3.2 The LS Algorithm

Our LS performs a descent algorithm (other techniques such as simulated annealing can be considered) in which we consider the first improving (w.r.t. the evaluation function) neighbour when moving. Diversification is achieved by restarting the algorithm (i.e., a loop around the algorithm of Figure 2). The result of the LS is thus a local optimum, the best configuration w.r.t. the evaluation function.

A configuration consists in giving a tentative value to each non instantiated variable of the CSP. The size of configurations thus changes at each LS since the number of non instantiated variables varies w.r.t. to enumerations, propagations, and backtracks. The algorithm is as generic as can be. It is parameterized by:

- $fMaxIter$ (resp. $fMaxRestart$), a function to compute the maximum number of iterations in a search (resp. the number of restarts i.e., of LS);
- $feval$, an evaluation function to estimate configurations,
- $fneighbour$, a function to compute neighbours of configurations.

We now present some of our functions (this list is not exhaustive, and some more functions can be designed). $fMaxIter$ and $fMaxRestart$ are functions based on the number of variables and constraints, e.g., functions such as $n.N_Var$ where n is a given integer and N_Var the current number of non instantiated variables.

The **evaluation function** is based on the constraints, the variables, their current domains, and/or their number of occurrences in the CSP. Thus, possible functions are $fConfC$ (the number of conflicting constraints), $fConfV$ (the number of conflicting variables), or functions such as $fCWeightff$ defined by $fCWeightff(V) = \sum_{i=1}^n 1/size(V_i)$ where V is a set of conflicting variables, and $size$ is a function that returns the number of values in the domain of a variable. Functions such as $fCWeightff$ give more importance to some variables

for evaluation (the ones with small domains for *fCWeightff*) in order to be combined with some variable selection strategies of the CP algorithm (e.g., a first-fail variable selection strategy in the case of *fCWeightff*). We thus have a set of evaluation functions to privilege (or un-privilege) the variables with the smallest domain, with the largest domain, the variables having more occurrences, ... We do not detail these other functions, their principle being similar to *fCWeightff*. These functions return 0 when the tentative CSP is solved.

The **neighbourhood functions** aim at finding an improving neighbour. A neighbour is computed by changing the tentative value of k variables. k is either a given integer, or the result of a function based on the number of variables (such as $\log(\text{Number_Variable})$). There are several criteria to select the variables to be changed: randomly, a randomly selected conflicting variable, and a family of functions based on the domain size and/or occurrences of variables (criteria similar to the criteria of the evaluation function). For this last function, a higher probability is given to the variables we want to privilege. The new tentative value is a randomly selected value of the domain of the variable.

Let us illustrate it for the *Nweightff2* neighbourhood function which gives more probability to change the value of a variable with a small domain. To this end, we construct a list of pairs $((V_1, I_1), (V_2, I_2), \dots, (V_n, I_n))$ where V_i are variables, and I_i are consecutive intervals of width $1/(\text{size}(V_i)^2)$ defined by:

$$I_1 = [0..(1/(\text{size}(V_1)^2))] \quad \text{and} \quad I_i = [\sum_{j=1}^{i-1} 1/(\text{size}(V_j)^2) .. \sum_{j=1}^i 1/(\text{size}(V_j)^2)]$$

Then, a random real number r between 0 and $\sum_{j=1}^n 1/(\text{size}(V_j)^2)$ is generated; the selected variable V_i is the one whose interval contains r , i.e., $r \in I_i$. Note that *NCweightff2* is defined the same way, but only considers conflicting variables. We do not detail the other neighbourhood functions based on smallest/largest domain, number of occurrences since they are built similarly to *Nweightff2*.

A LS strategy is thus a combination of the 4 parameters described above.

3.3 Hybrid Enumeration Strategies

Enumeration strategies are combinations of a variable selection criterion and a value selection criterion. We consider standard variable selection criteria of CP that we can refine using the result of the LS, e.g., selecting variables that are not conflicting in the LS. Here are some of the variable selection criteria:

- *first* (resp. *first_nc*) selects the first variable (resp. the first non conflicting variable) occurring in the CSP.
- *first-fail* (resp. *first-fail_nc*) selects the variable (resp. the non conflicting variable) with the smallest domain.
- *occurrence* (resp. *occurrence_nc*) selects the variable (resp. the non conflicting variable) with the largest number of occurrences in the set of constraints.

The value selection totally relies on the result of the LS: the tentative value (in the result of LS) of the variable is selected. An **hybrid strategy** is thus the

Table 1. n-queens problem with various strategies and hybrid strategies

	10 – queens		20 – queens		25 – queens		50 – queens	
	<i>t</i>	<i>e</i>	<i>t</i>	<i>e</i>	<i>t</i>	<i>e</i>	<i>t</i>	<i>e</i>
<i>first</i>	0.02	16	44.69	22221	13.05	4508	—	—
<i>S1</i>	0.051	15.01	0.473	51.57	0.937	73.31	12.17	702
<i>S4</i>	0.055	15.39	0.519	52.07	1.121	87.33	8.68	252.07
<i>ff</i>	0.02	13	0.09	40	0.20	68	2.78	506
<i>S6</i>	0.039	13.301	0.384	33.43	0.662	41.58	4.81	84.91
<i>S14</i>	0.043	13.60	0.385	33.23	0.639	37.51	5.41	114.14
<i>S15</i>	0.043	13.48	0.383	31.20	0.648	35.31	5.22	100.93
<i>S16</i>	0.04	13.00	0.401	33.44	0.624	32.64	4.39	65.76
<i>S23</i>	0.073	12.80	0.578	29.79	1.082	40.095	7.584	67.38
<i>S35</i>	0.084	12.56	0.576	25.66	1.045	33.56	7.822	77.08
<i>S36</i>	0.082	11.25	0.620	27.30	1.022	29.73	7.025	50.48
<i>S42</i>	0.101	7.85	0.790	15.51	1.427	20.26	10.627	33.78

combination of 6 parameters: the value and variable selection criteria, the functions to compute the numbers of iterations and restart, the evaluation function, and the neighbourhood function.

4 Experimental Results and Discussions

Our prototype implementation has been written with the ECLiPSe Constraint Programming System ⁴ using finite domain libraries for constraint propagation (propagation enforces generalized arc-consistency [1]) and the repair library which significantly eased the handling of tentative values in LS. We did not optimise the code. Instead, we kept it as open and parameterized as possible to allow various hybrid strategies. Similarly, we did not specialize the LS for specific problems, although it is well known that generic LS generally behaves poorly. The test were run on an Athlon 3000+ with 1Go of RAM. For each test we performed 1000 runs and we present the average.

Experimentations. Table 1 shows the results of several strategies for some instances of the n-queens problem. We use a standard model of the problem. The *alldifferent* global constraint is duplicated into the pair-wise differences of all variables in order to get a powerful pruning (specific propagator) whereas the LS can better count conflicts. We don't perform restart for LS (*MaxRestart* = 1) and the neighbourhood function changes the value of one variable ($k = 1$).

A column represents a problem instance; a row is the performance of a strategy for the different instances; *t* is the average CPU time in seconds and *e* the average number of enumerations (both good enumerations that lead to a solution, and bad ones that enforced backtrack).

We first compare strategies based on the *first* criterion: variables to enumerate are selected in the order of appearance in the CSP. *first* is the pure CP strategy:

⁴ We used ECLiPSe 5.3, an old version of ECLiPSe, since the future of ECLiPSe licensing is currently under negotiation. <http://eclipse.crosscoreop.com/eclipse/>

the smallest value of the domain of the selected variable is chosen for enumeration. S_1 and S_4 are hybrid strategies. S_1 is also based on a *first* criterion for variable selection; the selected value is the tentative value of the result of the LS like for each of our strategies. The evaluation function counts the number of conflicting constraints ($fConfC$). The neighbourhood changes a conflicting variable ($fNConf$). The size of the LS ($MaxIter$) is twice the number of variables in the problem. S_4 differs in that the first non conflicting variable is selected for enumeration (*first_nc*), and the evaluation function is the number of conflicting variables ($fConfV$). Both S_1 and S_4 show very good results compared to *first*, both in CPU time and enumerations. We don't have the results for 50-queens for the pure CP *first*: for 30-queens, more than 4.000.000 enumerations are required!

All the other strategies (*ff* and S_6 to S_{42}) are based on a *first-fail* variable criterion. *ff* is the pure CP strategy: enumeration with the smallest value of the domain of the variable with the smallest domain. S_6 and S_{23} also select the variable with smallest domain whereas $S_{14,15,16,35,36,42}$ select the non conflicting variable with the smallest domain. We can see that it tends to be better to select a non conflicting variable. The evaluation function for each $S_{14,15,16,35,36,42}$ is the number of violated constraints. With respect to other tests we performed for n-queens, it seems that using an evaluation function based on the size of the domain (such as *fCWeightff*) slightly reduces the number of enumerations but slightly increases the CPU time. For $S_{14,15,16,35,36,42}$, the number of iterations is $n.number_variables$, where $n = 2$ for $S_{14,15,16}$, $n = 5$ for $S_{23,35,36}$, and $n = 10$ for S_{42} . We can see that when n increases, the number of enumerations decreases, and the CPU time increases (due to the overhead of the longer LS). Increasing n is more profitable for larger problems, such as 100-queens.

In $S_{14,23,35}$, the neighbourhood function selects randomly a variable to change its value to create a neighbour; in $S_{6,15,36}$ it is a random conflicting variable whose value is changed; in S_{16} , *NCWeightff2* gives more probability to change the value of a variable with a small domain (this is coherent with the first-fail selection of variable for enumeration).

The difference between $S_{14,15,16}$ is the neighbourhood function: similarly to other tests we made, it tends that it is slightly worth (in terms of enumerations, and CPU time) having more clever neighbourhood functions based on conflicting variables and probabilities w.r.t. the size of the domains of the variables.

The advantage of the hybrid strategies over the pure CP *first-fail* strategy is in terms of enumerations: less enumerations are required to solve the problems. But this is at the cost of CPU time. However, the larger the instances the smaller the difference of time. Moreover, the overhead could be significantly reduced with a better implementation of the LS algorithm.

We test the same strategies on several instances of the Latin square problem. The comments are the same as for n-queens: all the hybrid strategies improve enumeration. The differences between the hybrid strategies are a bit more pronounced, but the main differences are coming from the length of the search. Only S_{42} behaves differently: it is from far the best strategy in terms of enumerations (e.g., less than half the number of enumerations for Latin-10), but also

the slowest one (2 to 3 times slower than other strategies for Latin-15). S_{42} is the strategy that performs the longest LS: this explains the good enumeration. But the overhead is not sufficient to explain the very slow run and we don't see yet any explanation for it.

Discussions. We were surprised that on average, all the hybrid strategies show quite similar performances. Only the length of the LS has a significant impact: the longer the LS, the less enumerations, but at the cost of CPU time. However, it seems that integrating criteria (e.g., domain size) in the neighbourhood and evaluation functions pays in term of enumeration and is not too costly in time.

As expected the hybrid strategies reduce the number of enumeration. This is due to 1) some better enumerations are found using a kind of probing (the LS), and 2) it also happens that the probing (the LS) finds a solution; most of the time, this phenomenon happens at the bottom of the search tree, when there remain few variables to be instantiated. However, the main reason is 1).

We did not measure the standard deviation and variance of the number of enumerations nor of the CPU time. However, we observed that the shorter is the LS, the larger is the variance. It could be interesting to study this deviation to compare hybrid strategies. Indeed, it can be more useful to have a strategy which is maybe a bit worse in average but that has a smaller deviation/variance; this way, runs would be more homogeneous.

In our experimental results, the difference (in terms of enumeration and time) between our hybrid strategies is less noticeable than between the pure CP strategies. For example, there are several orders of magnitude between the *first* and *first-fail* strategies in the n-queens problem. Moreover, our hybrid strategies are better (enumeration and time) than the pure *first* one, and close (better for enumerations, worse in time, but less than an order of magnitude) to the *first-fail* one. Thus, if one does not know which strategy is adapted to his problem, it is convenient to select an hybrid one: it will either gives good performance, or acceptable performance w.r.t. the best strategy. Moreover, the LS implementation can significantly be improved.

5 Conclusion, Related Work, and Future Work

We have presented an hybrid (and complete) solver that uses a LS to guide the enumeration process. Our technique is rather simple to integrate in a constraint programming system, and the first experimental results are rather promising.

In [10], a GSAT-like procedure guides the branching of logically-complete algorithms based on Davis and Putnam's like techniques. This could be seen as similar to our technique but in the context of SAT, and the heuristics and algorithms (both the complete and incomplete one) are different. [15] also reports about a repair-based algorithm, GSAT, combined with a constructive algorithm using propagation. At each node of the search tree, GSAT is run on all variables to choose which variable to enumerate. Our work is close to this algorithm since the systematic backtracking techniques are both enhanced with propagation. However, [15] addresses SAT problems, and thus the enumeration strategies, and

the incomplete techniques are different. From some aspects, in local probing [9], LS helps selecting variables. However, the key idea is that LS creates assignments, and CP modifies the sub-problem that LS is solving in order to guide it to search parts where solutions can be found. Although we did not observe it, more studies are needed to determine whether we avoid the heavy-tailed behaviour [7].

We plan to improve the implementation of our LS algorithm. We think of refining the notions of neighbourhood and evaluation to integrate some constraints of the model in these functions, i.e., to obtain a generic LS which can adapt itself to some specific constraints of the problem. We plan to extend our hybridisation to optimisation and perform some tests with other enumeration strategies.

References

1. K. R. Apt. *Principles of Constraint Programming*. Cambridge Univ. Press, 2003.
2. J. C. Beck, P. Prosser, and R. Wallace. Variable Ordering Heuristics Show Promise. In *Proc. of CP'2004*, volume 3258 of *LNCS*, pages 711–715, 2004.
3. Y. Caseau and F. Laburthe. Improved clp scheduling with task intervals. In *Proc. of ICLP'1994*, pages 369–383. MIT Press, 1994.
4. C. Castro and E. Monfroy. Designing hybrid cooperations with a component language for solving optimisation problems. In *Proceedings of AIMSA 2004*, volume 3192 of *LNCS*, pages 447–458. Springer, 2004.
5. F. Focacci, F. Laburthe, and A. Lodi. Local search and constraint programming. In *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research and Management Science*. Kluwer Academic Publishers, 2002.
6. H. Hoos and T. Stützle. *Stochastic Local Search: Foundations and Applications*. Morgan Kaufmann, San Francisco (CA), USA, 2004.
7. T. Hulubei and B. O'Sullivan. Search heuristics and heavy-tailed behaviour. In *Proceedings of CP'2005*, volume 3709 of *LNCS*, pages 328–342. Springer, 2005.
8. N. Jussien and O. Lhomme. Local search with constraint propagation and conflict-based heuristics. *Artif. Intell.*, 139(1):21–45, 2002.
9. O. Kamarainen and H. E. Sakkout. Local probing applied to network routing. In *CPAIOR*, volume 3011 of *LNCS*, pages 173–189. Springer, 2004.
10. B. Mazure, L. Sais, and É. Grégoire. Boosting complete techniques thanks to local search methods. *Ann. Math. Artif. Intell.*, 22(3-4):319–331, 1998.
11. E. Monfroy, F. Saubion, and T. Lambert. Hybrid csp solving. In *Proc. of FroCos 2005*, volume 3717 of *LNCS*, pages 138–167. Springer, 2005. Invited paper.
12. S. Prestwich. A hybrid search architecture applied to hard random 3-sat and low-autocorrelation binary sequences. In *Proceedings of CP'2000*, volume 1894 of *LNCS*, pages 337–352. Springer, 2000.
13. S. Prestwich. Local search and backtracking vs non-systematic backtracking. In *Proceedings of AAAI 2001 Fall Symposium on Using Uncertainty within Computation*, pages 109–115. AAAI Press, 2001. Technical Report FS-01-04.
14. M. Wallace. Hybrid algorithms, local search, and Eclipse. CP Summer School 05. <http://www.math.unipd.it/frossi/cp-school/wallace-lec-notes.pdf>, 2005.
15. M. Wallace and J. Schimpf. Finding the right hybrid algorithm - a combinatorial meta-problem. *Ann. Math. Artif. Intell.*, 34(4):259–269, 2002.

Study on Integrating Semantic Applications with Magpie

Martin Dzbor and Enrico Motta

Knowledge Media Institute, Computing Research Centre, The Open University, UK
{M.Dzbor, E.Motta}@open.ac.uk

Abstract. This paper describes two approaches to integrating standalone information processing techniques into a semantic application capable of acquiring and maintaining knowledge, which we conducted using our open Semantic Web framework of Magpie. We distinguish between integration through aggregation and through choreographing, and argue that the latter is not only simpler to realize but also provides greater benefits. The benefits were, in our experiment, related to developing a capability of maintaining and validating knowledge through an integration of down- and upstream knowledge processing tools. We describe the principles of integration and relate them to pragmatic challenges for the semantic web and to strategic directions of its evolution.

1 Introduction

Over the last two years we can observe an emphasis not only on designing and hand-crafting ontologies, but also on their automated population. This is arguably the key to bootstrapping the Semantic Web, and to making it suitable for practical, larger-scale applications. Current approaches to ontology population are based on various techniques; e.g. automated pattern recognition and extraction on the Web [5, 8] and web mining algorithms relying on information redundancy [2, 3]. These techniques are often considered as a part of ‘back-office support’ and are usually hidden from the users and user-facing interfaces.

The net effect of splitting the creation and population ontologies from their use and application is that the research into abstract infrastructures for the Semantic Web (e.g. [15]) and into specialized tools for expert knowledge engineers (e.g. [10]) is rather removed from the research into tools for the end users of the Semantic Web. These emerging tools support end users in annotating documents [13]; in browsing semantically marked up spaces [18], or in integrating web and semantic web resources [7]. They are characterized by *applying* rather than *developing* knowledge models, and their authors stress the importance of achieving some form of reward through good enough semantics rather than global completeness.

These developments signal that the Semantic Web research moves from its early vision and early adopter issues [1] to more pragmatic ones [14]. One of such pragmatic challenges is to re-use various specialized tools, techniques and methods when designing new semantically aware applications. In this paper we would like to explore the notion of re-use as an equivalent to an open infrastructure for facilitating interoperability between narrowly specialized modules and simultaneously, for bridging the gap between designing and applying ontologies. From a pragmatist’s and practitioner’s perspective the Semantic Web should take advantage of its distributed nature

and add value to practical applications *across a large part of the knowledge processing cycle*. By focusing on any one of the knowledge processing stages (e.g. annotation or formal representation), the likely practical benefits affect smaller user audiences. Thus, to overcome the well-known shortcomings of collaborative tools [11] that are remarkably applicable to the current version of the Semantic Web, we suggest focusing more on supporting interoperability in open, service-based infrastructures.

In this paper we discuss one particular approach taking on the strategic challenges of automated bootstrapping and supporting both knowledge engineers and end users. We look into how information extraction services and their combinations might interact directly with the end user. This interplay of ontology bootstrapping and user interaction issues has not been investigated so far. As a first step towards understanding requirements on integrating different knowledge tools we conducted a case study in which we extended a user-centered tool for browsing the Web using semantic relations (Magpie), and integrated it with tools for extracting information from visited text (C-PANKOW and Armadillo). Linking end user tools with the knowledge production tools is one approach to support the creation of semantic applications addressing a larger part of the knowledge processing cycle.

2 Making the Semantic Web More Practical

From the architectural point of view, there are several ways to realize such hybrid solutions. We consider two approaches to integrating the upstream and downstream knowledge tools: one focusing on data aggregation and another focusing on choreographing independent semantic services into a loose application that is capable of improving the scalability, bootstrapping and maintenance of knowledge, yet at the same time provides rewards to both users and knowledge engineers.

The requirement to support knowledge processing across a cycle rather than one particular phase arose from our earlier research into ontology-driven applications. One of the first usable tools with some Semantic Web functionality designed in early 2000-s was Magpie. Magpie comprises a service-based infrastructure and a browser plug-in that makes the ontological commitments accessible to the end user by rendering them into a visual form that is layered over a standard, non-semantic web page. The Magpie has evolved from the initial *handcrafted solution* that supported a fixed set of actions [6] to an *open solution*. This openness was achieved through user-selectable ontologies and dynamically definable, distributed services for navigating and reasoning on the Semantic Web [7]. However, both these solutions fell short of realizing the pragmatic challenges introduced earlier.

As Magpie, other applications fall short of addressing the knowledge processing chain in its entirety. For instance, Haystack [18] performs well on reuse and sharing but less well on acquisition and maintenance. Protégé [10] focuses on representing knowledge and ontology design, with basic versioning and mapping support. GATE [4] and its application in KIM [17] support discovery and annotation, but there is limited use, reuse and maintenance of the discovered knowledge. Web browsers (in general) and ontology browsers/visualizers (in particular) are focusing on knowledge presentation but not discovery, creation and maintenance.

We believe that in order to make the Semantic Web [7] more practical users need a toolkit that efficiently connects semantic and standard (i.e. non-semantic) browsing and navigating techniques, e.g. using the automated semantic annotation of web pages. However, the experience with Magpie shows that the merger of semantic and non-semantic is often brittle [20]. Magpie recognizes terms in a web page in real time with high precision whilst the page is *within* the user-selected ontological domain. When a Magpie with its lexicon is used outside the intended, well-defined domain, the performance of its automated annotation techniques rapidly falls.

One way to tackle brittleness is by extending the annotation techniques that rely on well-defined ontologies with a more open-ended form of knowledge discovery. This would allow *more robust browsers for the Semantic Web*, where robustness can be seen as a capability to recognize potentially relevant, not only well-defined knowledge. When we compare the brittle Magpie lexicons with those augmented by an information extraction (IE) tool (see also [20]), the users' performance in an exploratory task indeed improved when using the augmented lexicons where the domain boundaries were less brittle. However, this achievement came at the price of not being able to generate lexicons needed for the ontology-driven annotation in real time – knowledge discovery and validation simply take time.

Hence, the idea of robust browsing is clashing with real-time responsiveness. Nonetheless, if we reduce the problem into designing an 'interoperable shell' for building Semantic Web applications, the individual modular capabilities (e.g. ontology population using IE) can be fine-tuned without re-designing the entire application. Magpie makes a step towards such a shell: it provides generic mechanisms for integrating the ontologies, knowledge bases (KB) and web resources with the hyperlinks, (semantic) web services and tools interacting with them. Rather than producing applications for merely browsing the ontological structures of the Semantic Web, we propose to view the applications as *integrated, online solutions for processing knowledge (on the Semantic Web)*. This vision is novel, because it is an intersection of three research areas: Semantic Web, web services and knowledge management.

In our study we aimed to test the integration using Magpie as a shell based on two premises. First, we preferred turning existing, standalone IE tools into services that can be accessed through Magpie, rather than re-engineering Magpie's internal methods. Second, these independent services were loosely connected in a virtual framework from which a robust solution for some Semantic Web challenges may emerge. Our integration considers the needs of the end users (with their interest in browsing, retrieving and annotating), and of the knowledge engineers (who need to create and maintain KBs).

3 Motivation for an Integrated Application

As mentioned earlier, the adoption of Semantic Web depends on satisfying different needs of the different users. Focusing solely on end users is rewarding in the short term, but in addition to the instant, shallow and short-lived rewards, semantic applications should offer sustainable, "delayed gratification" [19].

As a scenario illustrating the potential value of such delayed reward, take a preparation of a review report. One starts with a few explicit keywords and domain

anchors. A large part of the review is about retrieving additional knowledge about the existing facts. In terms of a Magpie-based semantic application, semantic services supporting this task may include e.g. “Find papers on theme Y”. In addition to recalling and reusing the existing references, the review writing has a hidden but important *exploratory component* whereby we create new knowledge correlated with the existing facts. We want to create knowledge of e.g. “What has argument A in common with position P” or learn that “Topic T might be explained in terms of the discovered topic NT”. The state-of-the-art tools support either knowledge retrieval and *reuse*, or knowledge *creation*, but not both.

3.1 End User’s Perspective: Towards More Robust Knowledge Navigation

Fig. 1a shows a web page extract annotated by Magpie with a user-selected lexicon manually populated with several research activities. As can be expected, concepts like “Magpie” or “BuddySpace” are highlighted because they were explicitly defined in a KB. However, a few potentially relevant concepts are ignored (e.g. “agents” or “human language technology”). These are related to the existing terms and to the domain but are not in the KB. This incomplete lexicon contributes to the brittleness (i.e. rapid degradation of performance) of the ontology-based text annotation.

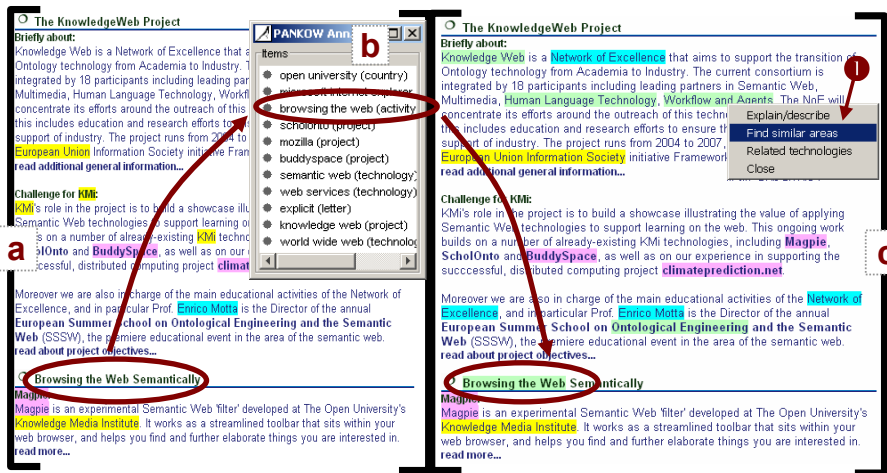


Fig. 1. Extract from a web page annotated with a brittle KB and lexicon (a), a list of KB extensions proposed by PANKOW and visualized in Magpie collector (b), and the same page annotated by an extended KB/lexicon as learned by PANKOW and validated by Armadillo (c)

To overcome brittleness, Fig. 1b shows a collection of additional concepts discovered by an offline IE tool and collected by Magpie’s dedicated interfaces. These terms relate to the ‘discourse’ of the user-selected lexicon, yet do not currently exist in the KB, from which annotation lexicons are generated. New terms are considered as instances, and our specific IE tool (which discovers these facts) also proposes a finer-grained classification of the discovered terms into existing classes such as “Activity”

or “*Technology*”. Fig. 1c shows some of the new terms turned into instances (e.g. “*browsing the web*” or “*workflow and agents*”) and made available to Magpie plug-in for an annotation. Importantly, items such as “*workflow and agents*” are not only highlighted as “*Research activities*”, but the user can also invoke associated services to obtain additional knowledge about these discoveries; e.g. the semantic menu marked as ❶ in Fig. 1c shows the user clicking on option “*Find similar areas*”, which in turn invokes a semantic service that uses correlation analysis to identify other research topics that co-occur with the discovered “*workflow and agents*” term.

3.2 Engineer’s Perspective: Towards Automated Knowledge Maintenance

The challenge of managing the scenario in section 3.1 is to avoid manual commitment of the discoveries to the ontology. Having an ontology/lexicon that evolves and exhibits learning and adaptive capabilities would be obviously beneficial to the user, but also to the knowledge engineer. From engineer’s perspective, the downside is the manual filtering, validation and inclusion of the discoveries from IE. In our experiments using C-PANKOW [2] as an arbitrary IE tool, the existing lexicon of 1,800 instances was extended by additional 273 organizations, 80 events, 176 technologies, etc. The engineer had to manually adjust classification for 21-35% of new items, which is still effort consuming.

This level of precision is to a great extent attributable to the algorithms used by the IE tool we used. While different IE tools may reduce the human intervention (e.g., IBM’s UIMA architecture and its IE engines [9] have a built-in functionality for relations and entity co-reference identification in addition to mere entity recognition), majority of research is dedicated to single-purpose, narrowly focused techniques. In other words, one needs to cope with the capabilities of the existing tools. In our study, we ‘coped’ by linking C-PANKOW to another (validating) IE technique, and this loose, dynamic link reduced the manual adjustment (e.g. for events) to 8%. Simply by making two different IE techniques interoperable and accessible from a semantic browser we achieved benefits in terms of evolving knowledge and assuring its quality.

This dynamic ontology population with an acceptable quality opens up further opportunities for knowledge engineers. For instance, ontologies can be populated so that they reflect personalized constraints on a generic KB. While the personalization was not the focus of this case study, it is an important side-effect of our strategic aim to develop a platform that is capable of addressing (at least in principle) a larger part of the knowledge processing cycle. Personalization, in turn, may lead to creating tools that are aware of such notions as trust or provenance [14], which offer an opportunity for e.g. social recommenders [12] and beyond.

4 Two Approaches to IE Tools Interoperability

Architecturally, there are many ways to realize the interoperability. Two approaches described below integrate different techniques, originally developed as standalone applications. Both use an existing, open semantic service based framework from Magpie [7]. From knowledge management perspective the two integration approaches can be

seen as combinations of different knowledge processing stages; they comprise knowledge acquisition, extension, validation, presentation and reuse.

As shown in Fig. 2, one knowledge processing cycle (starting by action 1 and ending with 8a) is shorter, shallower and rather trivial. It acts as a channel for delivering newly acquired and potentially relevant facts directly to the user: it integrates the facts *through aggregation* and is discussed in section 4.1. The other cycle (from 1 to 15, but replacing action 8a with 8b) is more complex. Unlike the former method, it relies on the interoperability of two (or more independent tools), which enables it to partially address the validity not only relevance of knowledge. This strategy facilitates *integration through choreographing* and is discussed in section 4.2.

4.1 Application Integration Through Aggregating Discovered Facts

Information aggregation is “a service that gathers relevant information from multiple sources” and “adds value by analyzing the aggregated information for specific objectives” [21]. Key capabilities of an aggregating service are: (i) to use multiple information sources, and (ii) to act as a central hub in the information exchange. Multiple providers of partial knowledge or information are typically unaware of each other and rely on the central hub – the aggregator – to mediate between them. Aggregating facts in one central service before distributing them to the users is akin to news feeds: every subscriber receives all news published in a given category. This lack of differentiation among the discovered (or published) facts is one of the limitations of this approach.

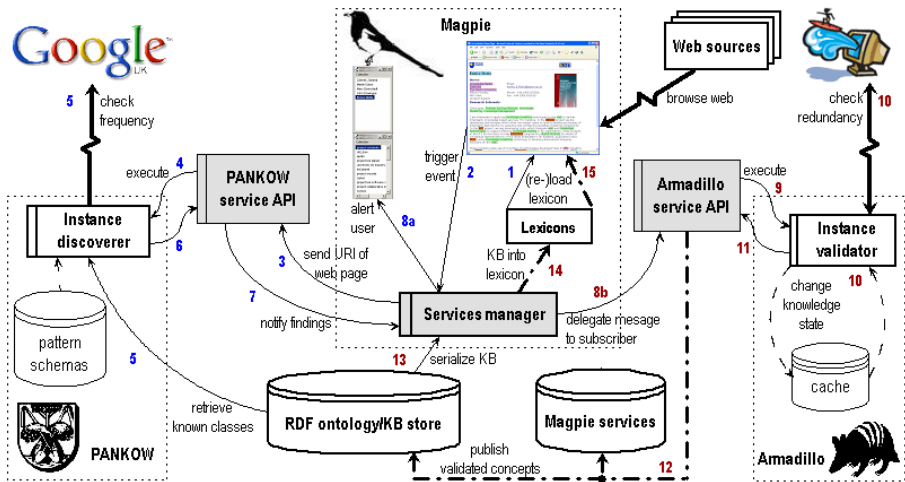


Fig. 2. Magpie – PANKOW – Amadillo interactions: *aggregation* (1-8a) and *choreography* (1-8b-15)

In our study we observed that knowledge maintenance was an asynchronous activity. For instance, C-PANKOW deployed techniques that drew on different complexity of search queries, and inevitably, the results from these sub-processes came at different times. The aggregator became a bottleneck – the user was not notified until all

facts were combined. This removed the repetition and allowed thresholding the discoveries based on a degree of confidence. Although, a tension arose between maintaining the real-time responsiveness of the semantic application [6] and the quality of discovered knowledge.

The human-computer interaction that emerges from aggregating the discovered instances and their subsequent presentation was purely user-driven. As shown in Fig. 2, this part of the integrative framework presented all (even trivial) discoveries to the user. In practice, knowledge delivery relied on generic visual user interfaces of the Magpie plug-in, which was not ideal if many new discoveries were made and reached the user all at the same moment.

The aggregation approach was realized by deploying the IE tool as a service, which was plugged into Magpie using its trigger services facility [6]. The IE service was triggered each time the user visited a new web page by an asynchronous XML-based invocation call. Upon completing information discovery, the user was alerted via a trigger service interface, where s/he saw a list with the discovered terms. These trigger interfaces allowed Magpie's standard semantic menus, if any were available. However, apart from classifying a new instance, no other properties were acquired.

The IE engine that has been published as a trigger service for Magpie and tested in the role of an aggregator was C-PANKOW [2]. The aggregation of IE discoveries in Magpie's collectors improved the original Magpie's capabilities; however, it was limited due to *mistakes in classifying* discoveries and due to *transient nature* of the IE service responses. Particularly the latter factor showed to be a major shortcoming of aggregation. This design constraint allowed alerting the user as close to the real time as possible, but at a price that the discoveries were not fully committed to the ontology in question. In fact, they were discarded at the end of each browsing session when Magpie was re-set. Knowledge that was not stored in lexicons/ontology before IE was lost for any future reuse. In practice, the value of discoveries is rarely obvious in isolation; the value comes from relating discovered chunks one to another.

4.2 Application Integration Through Choreographing IE Services

The second approach to integrating semantic tools assumes a reflection on results acquired by one tool/technique in another one. Instead of instant knowledge presentation typical for aggregation, the emphasis shifts to *knowledge validation*. To validate knowledge we need to go beyond mere discovery of additional facts in a corpus and their shallow classification [4]. In order to *create* new knowledge we need two consecutive steps: (i) an instantiation of the discovered concepts to appropriate classes, and (ii) shallow facts to be supported by relational knowledge.

Instead an instant user notification, the IE tool for validation is catching the output of the initial fact discovery step. Thus, additional services extending Magpie capabilities are not aggregated directly into the plug-in. Standalone services are instead connected in the background into a choreographed sequence. A choreographed process is defined by "the roles each party may play in the interaction; it tracks the sequence of messages that may involve multiple parties and multiple sources" [16]. Where aggregation assumes the information exchange between services is controlled by one party, choreography only tracks the message sequence, but no party owns the results of the conversation or information exchange [16].

Knowledge persistence of the outcomes from aggregated services is not a necessity, but in service choreographing it showed to be critical. Component services of a choreographed process had to be aware of the state the candidate chunk of knowledge is currently in – e.g. a discovered fact, classified individual, valid and extended knowledge are different states. Simple knowledge retrieval and aggregation services, such as described in [6] or in section 3.1 (see e.g. Fig. 1b), are state-less. The interaction is controlled by the aggregator. In case of scenario in Fig. 1b, the aggregator is equivalent to the specialized end user interface. By removing the aggregator as the owner of the conversation, we lose past discoveries and their classifications. Hence, the co-ordination among choreographed services needs to share knowledge states.

In our tests, C-PANKOW (again) was used as the first-pass IE service for discovering facts in a web page, but this time it delegated the results of the initial discovery to the Armadillo [3] service, that was put in a knowledge maintenance role. Armadillo was published into Magpie's open services framework as a special type of service that could be triggered by other services asserting particular patterns of the factual knowledge into a shared KB (so-called semantic log [6]) – rather than by the user's interaction with a web page as described e.g. in section 4.1. The semantic log maintained a persistent record of discovered facts and assertions about them (e.g. classification) in a local RDF store. The candidate facts from the store were checked against additional sources on the Web using the principle of information redundancy [3]:

- By proving the existence of certain 'expected' relationships we were able to *create new knowledge about the discovered facts*.
- The existence of the relationships using and extending the proposed class membership of an instance could be seen as *a form of IE validation*.

A discovery was validated when a new relational knowledge could be obtained using the ontological commitments made at an earlier stage. Once a fact is validated, it changes its state from a discovery to knowledge. It can be committed to the public repository containing the KB the original Magpie lexicon for annotating web pages was generated from. Magpie's lexicons are explicit serializations of KBs, and obviously, a different lexicon will be created from a changed KB. Magpie's support for loading a lexicon from a remote URI is beneficial for maintaining knowledge. When Armadillo validates PANKOW proposals and persistently publishes them in the RDF data store, the changes are propagated to the user next time a lexicon is requested from the KB.

5 Discussion

It might not be surprising to conclude from our tests that integration based on a loose choreography of independent services into a larger-scale and robust semantic application is the way for meeting pragmatic challenges for the Semantic Web. Indeed the nature of the Web and also the Semantic Web using such features as URI-addressable chunks of knowledge is fundamentally loose. However, more interesting outcome of our tests is that even fairly specialized applications (e.g. for information extraction or data classification) can be used in the context of Semantic Web applications. Moreover, these existing tools and techniques that are often *web-based* can be easily turned into services forming a part of a larger application.

What our extension of the Magpie framework with classic third-party IR techniques shows, is that the Semantic Web can be seamlessly bootstrapped from the existing (non-semantic) web applications. In fact, this form of linking together techniques specializing in different types of resources draws on the pluralistic notion of the Semantic Web being an open knowledge space with a variety of resources, which always mutually interact. If we want to highlight the primary benefit of choreography over aggregation, it is the aspect of *ownership*, *scalability* and an opportunity to bring in *social trust*. Because there is no need to design a new service overseeing the interaction of the component tools, this simplifies the application development. The notion of ‘overseeing’ is actually redundant to some extent – thanks to the functionality of asynchronous user-system interaction that underlies the Magpie framework. Significant time can thus be saved by the developers, who would otherwise need to re-design all tools involved in a typical, user-facing application with some semantic features (i.e. one needs equivalent functions of Magpie plug-in, C-PANKOW, Armadillo, etc.)

What the user receives in our integrated effort are candidate instances that pass several validating tests together with some of their properties and relations to other instances. These instances could be stored in a usual Semantic Web way; e.g. in a triple store. However, for the end user, these discovered instances and new knowledge is serialized into Magpie lexicons with a clear advantage that the user can interact with this new knowledge in a simple manner, without necessarily learning how to query semantic data stores and syntax of the data stored within.

In respect to knowledge validity, our choreographed integration improves on the individual component tools – mainly observable on the level of knowledge maintenance. While there are many techniques and strategies for knowledge discovery, representation and reuse, the maintenance of knowledge is in its infancy. *We are not aware of any major research into robust and scalable knowledge maintenance.* As has been mentioned earlier, one may try to add more functionalities to some of the existing tools, but this feature creep is not always beneficial. It often leads to bigger, more tightly coupled tools that make re-use more difficult.

Another important aspect is the capability of replicating creation of knowledge from both, direct experiences and social interactions. In our experiments, the direct experience is attributed to C-PANKOW drawing on the constructed hypotheses. The social creation came in a shape of Armadillo looking into trusted and established information sources for obtaining additional evidence for creating new knowledge. To generalize, this might be an approach to delivering a version of the semantic web, which is both *a formally and a socially constructed space*, and which contrasts with the current version, which mostly emphasizes formal, experience-based constructions.

Magpie framework is considerably simpler than many other products that to some extent act as semantic middleware. While no middleware is capable of solving such issues as real-time information extraction and maintenance on its own, the middleware (or what we referred to as a shell) may enable the application developers to build and/or replace modules more rapidly. The most recent example of the value brought by a lightweight interoperable framework can be seen in a semantic application designed by domain experts of an international advisory organization.

Knowledge acquisition was already in place in a form of various web services, web accessible dictionaries and vocabularies. Using the Magpie framework, the domain

experts (not the engineers or programmers!) created their first prototype of a semantic application by linking the appropriate existing services into Magpie accessible modules. Obviously, there is still a long way to go towards bringing the Magpie framework to the level of a fully reusable platform and middleware. Nevertheless, the current exploits of this technology help to show the potential benefits of semantic web technologies to the users who have not used them so-far.

One feature that emerged from this latest application of the Magpie framework is particularly suitable to conclude this paper: By linking the user with knowledge acquisition and maintenance, the application is actually *embedding* the Semantic Web features and technologies into the processes and activities the user normally carry out. Semantic relationships between the two or more concepts are not only visualized but, more practically, applied, used and exemplified in a concrete context of existing web resources. This embedding strategy makes it possible to achieve concrete benefit or a reward in a practical setting rather than artificially imposing an abstract structure of RDF graphs onto the user for whom this method of information processing is alien.

Acknowledgments

The effort described in this paper has been partially supported by the Dot.Kom, KnowledgeWeb, Advanced Knowledge Technologies (AKT) and NeOn projects. Dot.Kom, KnowledgeWeb and NeOn are sponsored by the European Commission by grants no. IST-2001-34038, FP6-507482 and FP6-027595, respectively. AKT is an Interdisciplinary Research Collaboration (IRC) sponsored by the UK Engineering and Physical Sciences Research Council by grant no. GR/N15764/01. I would like to acknowledge valuable technical and methodological inputs from Philipp Cimiano (University of Karlsruhe), Victoria Uren (The Open University) and Sam Chapman (University of Sheffield) to different aspects of work discussed in this paper, but particularly to the designs of different integration strategies.

References

- [1] Berners-Lee, T., Hendler, J., and Lassila, O., *The Semantic Web*. Scientific American, 2001. **279**(5): p. 34-43.
- [2] Cimiano, P., Ladwig, G., and Staab, S. *Gimme' the Context: Context-driven Automatic Semantic Annotation with C-PANKOW*. In *14th Intl. WWW Conf.* 2005. Japan.
- [3] Ciravegna, F., Dingli, A., Guthrie, D., *et al.* *Integrating Information to Bootstrap Information Extraction from Web Sites*. In *IJCAI Workshop on Information Integration on the Web*. 2003. Mexico.
- [4] Cunningham, H., Maynard, D., Bontcheva, K., *et al.* *GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications*. In *40th Anniversary Meeting of the Association for Computational Linguistics (ACL)*. 2002. Pennsylvania, US
- [5] Dill, S., Eiron, N., Gibson, D., *et al.* *SemTag and Seeker: bootstrapping the semantic web via automated semantic annotation*. In *Proc. of the 12th Intl. WWW Conf.* 2003. Hungary: ACM Press. p. 178-186.
- [6] Dzbor, M., Domingue, J., and Motta, E. *Magpie: Towards a Semantic Web Browser*. In *Proc. of the 2nd Intl. Semantic Web Conf.* 2003. Florida, USA. p. 690-705.

- [7] Dzbor, M., Motta, E., and Domingue, J. *Opening Up Magpie via Semantic Services*. In *Proc. of the 3rd Intl. Semantic Web Conf.* 2004. Japan. p. 635-649.
- [8] Etzioni, O., Cafarella, M., Downey, D., et al. *Methods for domain-independent information extraction from the web: An experimental comparison*. In *Proc. of the 19th AAAI Conf.* 2004. California, US. p. 391-398.
- [9] Ferrucci, D. and Lally, A., *Building an example application with the Unstructured Information Management Architecture*. IBM Systems Journal, 2004. **43**(3): p. 455-475.
- [10] Gennari, J., Musen, M.A., Fergerson, R., et al., *The evolution of Protege-2000: An environment for knowledge-based systems development*. Intl. Journal of Human-Computer Studies, 2003. **58**(1): p. 89-123.
- [11] Grudin, J., *Groupware and Social Dynamics: Eight Challenges for Developers*. Communications of the ACM, 1994. **37**(1): p. 92-105.
- [12] Heath, T., Motta, E., and Petre, M. *Person to Person Trust Factors in Word of Mouth Recommendation*. In *CHI2006 Workshop on Reinventing Trust, Collaboration, and Compliance in Social Systems (Reinvent06)*. 2006. Montreal, Canada.
- [13] Kahan, J., Koivunen, M.-R., Prud'Hommeaux, E., et al. *Annotea: An Open RDF Infrastructure for Shared Web Annotations*. In *10th Intl. WWW Conf.* 2001. Hong-Kong.
- [14] Kalfoglou, Y., Alani, H., Schorlemmer, M., et al. *On the Emergent Semantic Web and Overlooked Issues*. In *Proc. of the 3rd Intl. Semantic Web Conf.* 2004. Japan. p. 576-590.
- [15] Motik, B. and Sattler, U. *Practical DL Reasoning over Large ABoxes with KAON2*. 2006.
- [16] Peltz, C., *Web Services Orchestration & Choreography*. Web Services Journal, 2003,**3**(7)
- [17] Popov, B., Kiryakov, A., Kirilov, A., et al. *KIM - Semantic Annotation Platform*. In *Proc. of the 2nd Intl. Semantic Web Conf.* 2003. Florida, USA. p. 834-849.
- [18] Quan, D., Huynh, D., and Karger, D.R. *Haystack: A Platform for Authoring End User Semantic Web Applications*. In *Proc. of the 2nd Intl. Semantic Web Conf.* 2003. Florida, USA. p. 738-753.
- [19] Takeda, H. and Ohmukai, I. *Building semantic web applications as information sharing systems*. In *UserSWeb: Wksp. on User Aspects of the Semantic Web*. 2005. Crete.
- [20] Uren, V.S., Cimiano, P., Motta, E., et al. *Browsing for Information by Highlighting Automatically Generated Annotations: User Study and Evaluation*. In *Proc. of the 3rd Knowledge Capture Conf.* 2005. Canada. p. 75-82.
- [21] Zhu, H., Siegel, M.D., and Madnick, S.E. *Information Aggregation – A Value-added e-Service*. In *Proc. of the Intl. Conference on Technology, Policy and Innovation*. 2001. The Netherlands.

N-Gram Feature Selection for Authorship Identification

John Houvardas and Efstathios Stamatatos

Dept. of Information and Communication Systems Eng.
University of the Aegean
83200 – Karlovassi, Greece
stamatatos@aegean.gr

Abstract. Automatic authorship identification offers a valuable tool for supporting crime investigation and security. It can be seen as a multi-class, single-label text categorization task. Character n-grams are a very successful approach to represent text for stylistic purposes since they are able to capture nuances in lexical, syntactical, and structural level. So far, character n-grams of fixed length have been used for authorship identification. In this paper, we propose a variable-length n-gram approach inspired by previous work for selecting variable-length word sequences. Using a subset of the new Reuters corpus, consisting of texts on the same topic by 50 different authors, we show that the proposed approach is at least as effective as information gain for selecting the most significant n-grams although the feature sets produced by the two methods have few common members. Moreover, we explore the significance of digits for distinguishing between authors showing that an increase in performance can be achieved using simple text pre-processing.

1 Introduction

Since early work on 19th century, authorship analysis has been viewed as a tool for answering literary questions on works of disputed or unknown authorship. The first computer-assisted approach aimed at solving the famous *Federalist Papers* case [1] (a collection of essays, a subset of which claimed by both Alexander Hamilton and James Madison). However, in certain cases, the results of authorship attribution studies on literary works were considered controversial [2]. In recent years, researchers have paid increasing attention to authorship analysis in the framework of practical applications, such as verifying the authorship of emails and electronic messages [3,4], plagiarism detection [5], and forensic cases [6].

Authorship identification is the task of predicting the most likely author of a text given a predefined set of candidate authors and a number of text samples per author of undisputed authorship [7, 8]. From a machine learning point of view, this task can be seen as a single-label multi-class text categorization problem [9] where the candidate authors play the role of the classes.

One major subtask of the authorship identification problem is the extraction of the most appropriate features for representing the style of an author, the so-called *stylometry*. Several measures have been proposed, including attempts to quantify vocabulary richness, function word frequencies and part-of-speech frequencies. A good review of stylometric techniques is given by Holmes [10]. The vast majority of

proposed approaches are based on the fact that a text is a sequence of words. A promising alternative text representation technique for stylistic purposes makes use of character n -grams (contiguous characters of fixed length) [11, 12]. Character n -grams are able to capture complicated stylistic information on the lexical, syntactic, or structural level. For example, the most frequent character 3-grams of an English corpus indicate lexical (l_{thel}¹, l_{tol}, l_{thal}), word-class (ling_l, led_l), or punctuation usage (l_l__{Tl}, l_l“_{Tl}) information. Character n -grams have been proved to be quite effective for author identification problems. Keselj et al. [12] tested this approach in various test collections of English, Greek, and Chinese text, improving previously reported results. Moreover, a variation of their method achieved the best results in the ad-hoc authorship attribution contest [13], a competition based on a collection of 13 text corpora in various languages (English, French, Latin, Dutch, and Serbian-Slavonic). The performance of the character n -gram approach was remarkable especially in cases with multiple candidate authors (>5).

Tokenization is not needed when extracting character n -grams, thus making the approach language independent. On the other hand, they considerably increase the dimensionality of the problem in comparison to word-based approaches. Due to this fact, n -grams of fixed length have been used so far (e.g. 3-grams). The selection of an optimal n depends on the language. Dimensionality reduction is of crucial importance, especially in case we aim to extract variable-length n -grams. That is, the combination of all 2-grams, 3-grams, 4-grams, etc. is much higher than the word-forms found in a text. Therefore when variable length n -grams are used, an aggressive feature selection method has to be employed to reduce the dimensionality of the feature space. To this end, traditional feature selection methods, such as information gain, chi square, mutual information etc. could be used. In general, these methods consider each feature independent of the others and attempt to measure their individual significance for class discrimination.

In this paper, we propose a feature selection method for variable-length n -grams based on a different view. The original idea is based on previous work for extracting multiword terms (word n -grams of variable length) from texts in the framework of information retrieval applications [14, 15]. According to the proposed approach, each feature is compared with other similar features of the feature set and the most important of them is kept. The factor that affects feature importance is its frequency of occurrence in the texts rather than its ability to distinguish between classes. Therefore, following the proposed method, we produce a feature subset which is quite different with the one produced by a traditional feature selection method. Experiments on a subset of the new Reuters corpus show that our approach is at least as effective as information gain for distinguishing among 50 authors when a large initial feature set is used while it is superior for small feature sets. Moreover, we examine a simple pre-processing procedure for removing redundancy in digits found in texts. It is shown that this procedure improves the performance of the proposed approach.

The rest of this paper is organized as follows. Section 2 presents our approach for n -gram feature selection. Section 3 presents the corpus used in the experiments and a baseline method. Section 4 includes the performed experiments while in section 5 the conclusions drawn by this study are summarized and future work directions are given.

¹ We use ‘l’ and ‘_’ to denote n -gram boundaries and a single space character, respectively.

2 N-Gram Feature Selection

The proposed method for variable-length n-gram feature selection is based on an existing approach for extracting multiword terms (i.e., word n-grams of variable length) from texts. The original approach aimed at information retrieval applications (Silva [15]). In this study, we slightly modified this approach in order to apply it to character n-grams for authorship identification. The main idea is to compare each n-gram with similar n-grams (either longer or shorter) and keep the dominant n-grams. Therefore, we need a function able to express the “glue” that sticks the characters together within an n-gram. For example, the “glue” of the n-gram |the_| will be higher than the “glue” of the n-gram |theal.

2.1 Selecting the Dominant N-Grams

To extract the dominant character n-grams in a corpus we modified the algorithm *LocalMaxs* introduced in [15]. It is an algorithm that computes local maxima comparing each n-gram with similar n-grams. Given that:

- $g(C)$ is the *glue* of n-gram C , that is the power holding its characters together.
- $ant(C)$ is an *antecedent* of an n-gram C , that is a shorter string having size $n-1$.
- $succ(C)$ is a *successor* of C , that is, a longer string of size $n+1$, i.e., having one extra character either on the left or right side of C .

Then, the dominant n-grams are selected according to the following rules:

$$\begin{aligned}
 & \text{if } (C.length > 3) \\
 & \quad g(C) \geq g(ant(C)) \wedge g(C) > g(succ(C)), \forall ant(C), succ(C) \\
 & \text{if } (C.length = 3) \\
 & \quad g(C) > g(succ(C)), \forall succ(C)
 \end{aligned} \tag{1}$$

In the framework of authorship identification task, we only consider 3-grams, 4-grams, and 5-grams as candidate n-grams, since previous studies have shown they provide the best results [12]. As an alternative, we also consider words longer than 5 characters as candidate n-grams. Note that 3-grams are only compared with successor n-grams. Moreover, in case no words are used, 5-grams are only compared with antecedent n-grams. So, it is expected that the proposed algorithm will favor 3-grams and 5-grams against 4-grams.

2.2 Representing the Glue

To measure the glue holding the characters of a n-gram together various measures have been proposed, including specific mutual information [16], the ϕ^2 measure [17], etc. In this study, we adopt the *Symmetrical Conditional Probability* (SCP) proposed in [14]. The SCP of a bigram |xy| is the product of the conditional probabilities of each given the other:

$$SCP(x, y) = p(x|y) \cdot p(y|x) = \frac{p(x, y)}{p(x)} \cdot \frac{p(x, y)}{p(y)} = \frac{p(x, y)^2}{p(x) \cdot p(y)} \quad (2)$$

Given a character n -gram $|c_1 \dots c_n|$, a *dispersion point* defines two subparts of the n -gram. A n -gram of length n contains $n-1$ possible dispersion points (e.g., if $*$ denote a dispersion point, then the 3-gram $|lhel|$ has two dispersion points: $|l*hel|$ and $|lth*el|$). Then, the SCP of the n -gram $|c_1 \dots c_n|$ given the dispersion point $|c_1 \dots c_{n-1} * c_n|$ is:

$$SCP((c_1 \dots c_{n-1}), c_n) = \frac{p(c_1 \dots c_n)^2}{p(c_1 \dots c_{n-1}) \cdot p(c_n)} \quad (3)$$

The SCP measure can be easily extended so that to account for any possible dispersion point (since this measure is based on fair dispersion point normalization, will be called *fairSCP*). Hence the *fairSCP* of the n -gram $|c_1 \dots c_n|$ is as follows:

$$fairSCP(c_1 \dots c_n) = \frac{p(c_1 \dots c_n)^2}{\frac{1}{n-1} \sum_{i=1}^{i=n-1} p(c_1 \dots c_i) \cdot p(c_{i+1} \dots c_n)} \quad (4)$$

3 Experimental Settings

3.1 Corpus

In 2000, a large corpus for the English language, the Reuters Corpus Volume 1 (RCV1) including over 800,000 newswire stories, become available for research purposes. A natural application of this corpus is to be used as test bed for topic-based text categorization tasks [18] since each document has been manually classified into a series of topic codes (together with industry codes and region codes). There are four main topic classes: CCAT (corporate/industrial), ECAT (economics), GCAT (government/social), and MCAT (markets). Each of these main topics has many subtopics and a document may belong to a subset of these subtopics. Although, not particularly designed for evaluating author identification approaches, the RCV1 corpus contains ‘by-lines’ in many documents indicating authorship. In particular, there are 109,433 texts with indicated authorship and 2,361 different authors in total.

RCV1 texts are short (approximately 2KBytes – 8KBytes), so they resemble a real-world author identification task where only short text samples per author may be available. Moreover, all the texts belong to the same text genre (newswire stories), so the genre factor is reduced in distinguishing among the texts. On the other hand, there are many duplicates (exactly the same or plagiarized texts). The application of R -measure to the RCV1 text samples has revealed a list of 27,754 duplicates [19].

The RCV1 corpus has already been used in author identification experiments. In [19] the top 50 authors (with respect to total size of articles) were selected. Moreover, in the framework of the *AuthorID* project, the top 114 authors of RCV1 with at least 200 available text samples were selected [20]. In contrast to these approaches, in this study, the criterion for selecting the authors was the topic of the available text samples. Hence, after removing all duplicate texts found using the R -measure, the top

50 authors of texts labeled with at least one subtopic of the class CCAT (corporate/industrial) were selected. That way, it is attempted to minimize the topic factor in distinguishing among the texts. Therefore, since steps to reduce the impact of genre have been taken, it is to be hoped that authorship differences will be a more significant factor in differentiating the texts. Consequently, it is more difficult to distinguish among authors when all the text samples deal with similar topics rather than when some authors deal mainly with economics, others with foreign affairs etc. The training corpus consists of 2,500 texts (50 per author) and the test corpus includes other 2,500 texts (50 per author) non-overlapping with the training texts.

3.2 Information Gain as Baseline

Most traditional feature selection methods are information-theoretic functions attempting to measure the significance of each feature in distinguishing between the classes. In [21] the main feature selection methods are extensively tested in the framework of (topic-based) text categorization experiments. Among document frequency thresholding, information gain, mutual information, chi square, and term strength, the most effective methods were proved to be information gain and mutual information.

Information gain represents the entropy reduction given a certain feature, that is, the number of bits of information gained about the category by knowing the presence or absence of a term in a document:

$$IG(t_k, c_i) = \sum_{c \in \{c_1, \dots, c_i\}} \sum_{t \in \{t_k, \dots, t_k\}} P(t, c) \cdot \log \frac{P(t, c)}{P(t) \cdot P(c)} \quad (5)$$

Since information gain considers each feature independent of the others, it is not able to detect multiple redundant features that have the same ability to distinguish between classes. On the other hand, it offers a ranking of the features according to

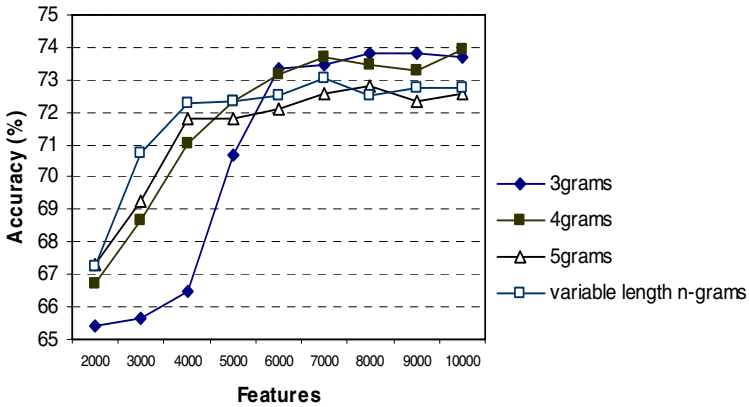


Fig. 1. Authorship identification results using information gain for feature selection

their information gain score, so a certain number of features can be easily selected. In this study, information gain was used as the baseline feature selection method. Any proposed method should have performance at least equal with the performance of information gain.

3.3 Author Identification Experiments

In each performed experiment, the following procedure was followed:

- An initial large feature set consisting of n -grams of variable length is extracted from the training corpus. This feature set includes the L most frequent n -grams for certain values of n . That is, for $L=5,000$, the 5,000 most frequent 3-grams, the 5,000 most frequent 4-grams, and the 5,000 most frequent 5-grams compose the initial feature set. In some cases, the most frequent long words ($\text{length} > 5$) are also added to the initial feature set.
- A feature selection method is applied to this large feature set.
- A Support Vector Machine (SVM) is trained using the reduced feature set. In all experiments, linear kernels are used with $C=1$.
- The SVM model is applied to the test set and the microaverage accuracy is calculated.

4 Results

The first experiment was based on information gain measure to select the most significant features. Given an initial feature set of 15,000 features (including the 5,000 most frequent 3-grams, the 5,000 most frequent 4-grams, and the 5,000 most frequent 5-grams) information gain was used to extract the best 2,000 to 10,000 features with a step of 1,000. For comparative purposes, we also used information gain to select fixed-length n -grams. Hence, using as initial feature set the 15,000 most frequent 3-grams, information gain was used to select the best 2,000 to 10,000 features with a step of 1,000. The same approach was followed for 4-grams and 5-grams. The results are shown in Figure 1. As can be seen, the variable-length n -grams outperform fixed-length n -grams for relatively low dimensionality (when less than 5,000 features are selected). However, when the dimensionality arises, the variable-length n -grams selected by information gain fail to compete with fixed-length n -grams (especially, 3-grams and 4-grams). Moreover, in all cases the performance of the model is not significantly improved beyond a certain amount of selected features.

In the second experiment, we applied the proposed method to the same problem. Recall that our method is not able to select a predefined number of features since it does not provide feature ranking. However, the number of selected features depends on the size of the initial feature set. So, different initial feature sets were used, including 6,000 to 24,000 variable-length n -grams, equally distributed among 3-grams, 4-grams, and 5-grams. Moreover, we also performed experiments using words longer than 5 characters as candidate n -grams. The results of these experiments are depicted in Figure 2. Note that the results of this figure are not directly comparable with the results of figure 1 since different initial feature sets were used. When using exactly the same initial feature set with information gain, the accuracy based on our

method reaches 73.08%. It can be seen that the proposed method can produce much more accurate classifiers in comparison with information gain when using a low number of features. In addition, these reduced feature sets were selected from a more restricted initial feature set. For example, when the initial feature set comprises 6,000 variable-length n-grams, our method selects 2,314 features producing an accuracy of 72%. Recall that a feature set of 4,000 variable length n-grams (selected out of 15,000 n-grams) produced by information gain reaches accuracy of 72%. On the other hand, the addition of long words to the feature set does not seem to significantly contribute to the classification accuracy.

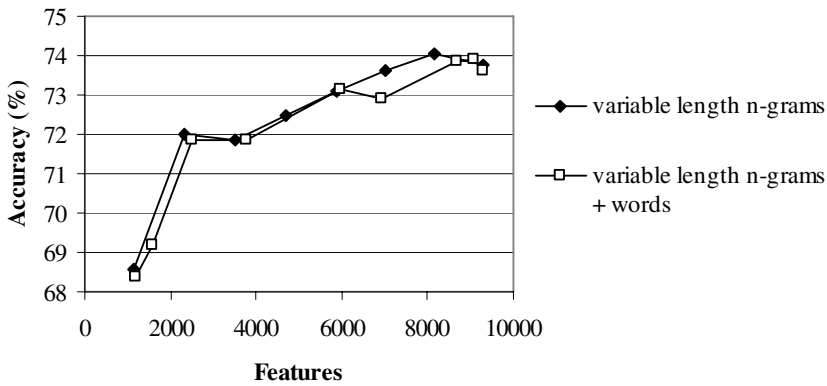


Fig. 2. Results of the proposed method using only variable-length n-grams and variable-length n-grams plus words longer than 5 characters

Table 1. Comparison of the feature sets produced by information gain (IG) and the proposed method (PM) in terms of common members (CM) for three cases

	IG	PM	CM	IG	PM	CM	IG	PM	CM
3-grams	647	1337	127	647	2,938	317	851	5,510	530
4-grams	909	423	161	2,228	705	462	2,327	1,012	510
5-grams	758	554	131	1,816	1,048	315	5,000	1,656	1,257
Total	2,314	2,314	419	4,691	4,691	1094	8,178	8,178	2,297
Accuracy	69.4%	72.00%		72.16%	72.48%		72.56%	74.04%	

A closer look at the feature sets produced by information gain and the proposed method will reveal their properties. To this end, table 1 presents the distribution in 3-grams, 4-grams, and 5-grams of the best features produced by information gain and the proposed method, respectively, as well as the amount of common members of the two sets. Three cases are shown corresponding to 2,314, 4,691, and 8,178 best features selected by the two methods. As can be seen, information gain favors 4-grams and especially 5-grams for large feature sets while for small feature sets the selected features are (roughly) equally distributed. On the other hand, the proposed method mainly favors 3-grams in all cases, followed by 5-grams. Interestingly, the common members of the two datasets are only a few. For example, in case of 2,314

best features, the proposed method selected 1,337 3-grams and the information gain selected 647 3-grams. However, the intersection of the two sets consists of 127 3-grams only. This indicates that the examined methods focus on different kinds of information when selecting the features. Indeed, information gain will select all the n-grams `landl`, `land_`, `_andl`, `_and_` given that the use of word ‘and’ is important for distinguishing between the authors. Thus, the reduced feature set will contain redundant features. On the other hand, the proposed method will select at least one of these n-grams. Hence, when equal number of features is selected by the two methods, the feature set of the proposed method will be richer in different n-grams corresponding to different kind of stylistic information.

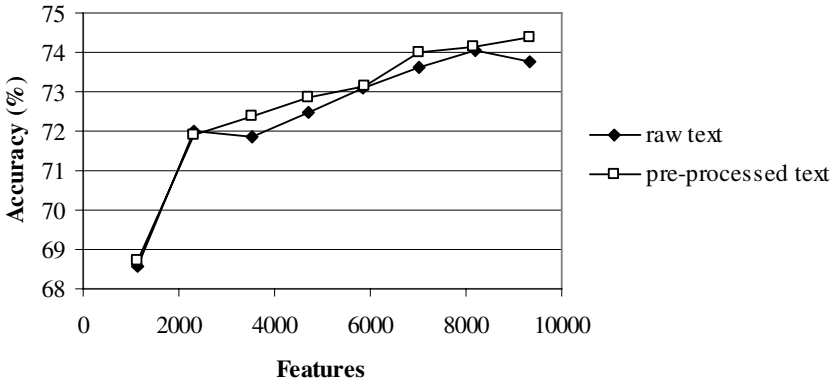


Fig. 3. Performance results using raw text and pre-processed text where all digit characters were replaced by the same symbol

4.1 Text Pre-processing

The experiments we have presented so far were conducted on raw text. No pre-processing of the text was performed apart from removing XML tags irrelevant to the text itself. However, simple text pre-processing may have a considerable impact in the framework of text categorization tasks [22]. In this study, we emphasize on pre-processing texts for removing redundancy of digit characters.

The information represented by digits may correspond to dates, values, telephone numbers etc. The use of digits is mainly associated with text-genre (press reportage, press editorial, official documents, etc.) rather than authorship. Given a character n-gram text representation, multiple digit-based n-grams will be extracted from a text. In many cases, the important stylistic information is the use of digits rather than the exact combinations of digits. Hence, if all digits are replaced with a special symbol (e.g., ‘@’), the redundancy in character n-grams would be much lower. For example, all `|1999|`, `|2000|`, `|2001|`, and `|2002|` 4-grams would be replaced by `|@@@@|`. Frequent use of this transformed 4-gram could be due to frequent reference to dates.

We examine the effect of this simple pre-processing procedure on the authorship identification task. Figure 3 depicts the classification accuracy results using the proposed feature selection method on variable-length n-grams extracted from raw text

(as previously) and pre-processed text (with digit characters replaced by a symbol). The amount of features selected based on the pre-processed text is slightly smaller. More importantly, the performance of the model based on pre-processed text is better especially when using more than 2,000 features. This indicates that simple text transformations can yield considerable improvement in accuracy.

5 Discussion

We presented a new approach for feature selection aimed at authorship identification based on character n-gram text representation. The proposed method is able to select variable length n-grams based on a technique originally applied for extracting multiword expressions from text. The key difference with traditional feature selection methods is that the significance of a feature is measured in comparison with other similar features rather than its individual ability to discriminate between the classes. Therefore, the produced feature set is stylistically richer since it contains the dominant character n-grams and is less likely to be biased by some powerful n-grams that essentially represent the same stylistic information.

Another difference with traditional feature selection approaches is that there is no ranking of the features according to their significance. Essentially, it is not possible to select a predefined number of features. Although this fact complicates the experimental comparison with other approaches, it is not of crucial importance for the practical application of the proposed method to real-world cases.

We also presented experiments about the significance of digits in the framework of author identification tasks. The removal of redundancy in digit characters improves classification accuracy when a character n-gram text representation is used. Furthermore, the cost of this procedure is trivial. It remains to be tested whether alternative text transformations are useful as well.

In this study, we restricted our method to certain n-gram types (3-grams, 4-grams, and 5-grams). To keep dimensionality on low level, we used words longer than 5 characters as an alternative for longer n-grams. However, the results when using the additional words were not encouraging. It would be interesting for one to explore the full use of long n-grams as well as the distribution of selected n-grams into different n-gram lengths especially when texts from different natural languages are tested.

References

1. Mosteller, F., Wallace, D.: Inference in an Authorship Problem. *Journal of the American Statistical Association*, 58:302 (1963) 275-30.
2. Labbé, C., Labbé, D.: Inter-textual distance and authorship attribution: Corneille and Molière. *Journal of Quantitative Linguistics*, 8 (2001) 213-31.
3. de Vel, O., Anderson, A., Corney, M., Mohay, G.: Mining E-mail Content for Author Identification Forensics. *SIGMOD Record*, 30:4 (2001) 55-64.
4. Abbasi, A., Chen, H.: Applying Authorship Analysis to Extremist-Group Web Forum Messages. *IEEE Intelligent Systems*, 20:5 (2005) 67-75.
5. van Halteren, H.: Linguistic Profiling for Author Recognition and Verification. In *Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics* (2004) 199-206.

6. Chaski, C.: Empirical Evaluations of Language-based Author Identification Techniques. *Forensic Linguistics*, 8:1 (2001) 1-65.
7. Stamatatos, E., Fakotakis, N., Kokkinakis, G.: Automatic Text Categorization in Terms of Genre and Author. *Computational Linguistics* 26:4 (2000) 471-495.
8. Peng, F., Shuurmans, F., Keselj, V., Wang, S.: Language Independent Authorship Attribution Using Character Level Language Models. In *Proc. of the 10th European Association for Computational Linguistics* (2003).
9. Sebastiani, F.: Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34:1 (2002) 1-47.
10. Holmes, D.: The Evolution of Stylometry in Humanities Scholarship. *Literary and Linguistic Computing*, 13:3 (1998) 111-117.
11. Kjell, B., Addison Woods, W., Frieder O.: Discrimination of authorship using visualization. *Information Processing and Management* 30:1 (1994).
12. Keselj, V., Peng, F., Cercone, N. Thomas, C.: N-gram-based Author Profiles for Authorship Attribution. In *Proc. of the Conference Pacific Association for Computational Linguistics* (2003).
13. Juola, P.: Ad-hoc Authorship Attribution Competition. In *Proc. of the Joint ALLC/ACH2004 Conf.* (2004) 175-176.
14. Silva, J., Dias, G., Guillore S., Lopes, G.: Using LocalMaxs Algorithm for the Extraction of Contiguous and Non-contiguous Multiword Lexical Units. *LNAI*, 1695 (1999) 113-132.
15. Silva, J., Lopes, G.: A local Maxima Method and a Fair Dispersion Normalization for Extracting Multiword Units. In *Proc. of the 6th Meeting on the Mathematics of Language* (1999) 369-381.
16. Church K., Hanks K.: Word Association Norms, Mutual Information and Lexicography. *Computational Linguistics*, 16:1 (1990) 22-29.
17. Gale W., Church K.: Concordance for parallel texts. In *Proc. of the 7th Annual Conference for the new OED and Text Research*, Oxford (1991) 40-62.
18. Lewis, D., Yang, Y., Rose, T., Li, F.: RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research*, 5 (2004) 361-397.
19. Khmelev, D. Teahan, W.: A Repetition Based Measure for Verification of Text Collections and for Text Categorization. In *Proc. of the 26th ACM SIGIR* (2003) 104-110.
20. Madigan, D., Genkin, A., Lewis, D., Argamon, S., Fradkin, D., Ye, L.: Author Identification on the Large Scale. In *Proc. of CSNA* (2005).
21. Yang, Y., Pedersen J.: A Comparative Study on Feature Selection in Text Categorization. In *Proc. of the 14th Int. Conf. on Machine Learning* (1997).
22. Marton, Y., Wu, N., Hellerstein, L.: On Compression-Based Text Classification. *Advances in Information Retrieval: 27th European Conference on IR Research*, Springer LNCS – 3408, pp. 300-314 (2005).

Incorporating Privacy Concerns in Data Mining on Distributed Data

Hui-zhang Shen¹, Ji-di Zhao¹, and Ruipu Yao²

¹ Aetna School of Management, Shanghai Jiao Tong University, Shanghai, P.R. China, 200052
{hzshen, judyzhao33}@sjtu.edu.cn

² School of Information Engineering, Tianjin University of Commerce, Tianjin, P.R. China, 300134
Yaoruipu@yahoo.com.cn

Abstract. Data mining, with its objective to efficiently discover valuable and inherent information from large databases, is particularly sensitive to misuse. Therefore an interesting new direction for data mining research is the development of techniques that incorporate privacy concerns and to develop accurate models without access to precise information in individual data records. The difficulty lies in the fact that the two metrics for evaluating privacy preserving data mining methods: privacy and accuracy are typically contradictory in nature. We address privacy preserving mining on distributed data in this paper and present an algorithm, based on the combination of probabilistic approach and cryptographic approach, to protect high privacy of individual information and at the same time acquire a high level of accuracy in the mining result.

1 Introduction

The issue of maintaining privacy in data mining has attracted considerable attention over the last few years. Previous work can be broadly classified into cryptographic approach and probabilistic approach. In the first approach, privacy preserving is achieved using cryptographic methods. [1] addressed secure mining of association rules over horizontally partitioned data. Their method incorporate cryptographic techniques and random values to minimize the information shared, thus the weakness of this method exists in the leakage of the encryption key and the random value. The basic idea in probabilistic approach is to modify data values such that reconstruction of the values for any individual transaction based on the distorted data is difficult and thus is safe to use for mining, while on the other hand, the distorted data and information on the distribution of the random data used to distort the data, can be used to generate valid rules and simulate an approximation to the original data distribution. Randomization is done using a statistical method of value distortion [6] that returns a value $x_i + r$ instead of x_i , where r is a random value drawn from a specific distribution. A Bayesian procedure for correcting perturbed distributions is proposed and three algorithms for building accurate decision trees that rely on reconstructed distributions are presented in [7] and [8]. [4] studied the feasibility of building accurate classification models using training data in which the sensitive numeric

values in a user's record have been randomized so that the true values cannot be estimated with sufficient precision. More Recently, the data distortion method has been applied to Boolean association rules [9][10]. [10] investigate whether users can be encouraged to provide correct information by ensuring that the mining process cannot violate their privacy on some degree and put forward an algorithm named MASK(Mining Associations with Secrecy Konstraints) based on Bernoulli probability model. [9] presented a framework for mining association rules from transactions consisting of categorical items and proposed a class of randomization operators for maintaining data privacy.

In this paper, we incorporate the cryptographic approach with the probabilistic approach, and address the problem of mining association rules in the context of distributed database environments, that is, all sites have the same schema, but each site has information on different entities. Thus a classical association mining rule such as Apriori should be extended to distributed environments to generate association rules. We assume such a scenario: a transaction database DB is horizontally partitioned among n sites which are S_1, S_2, \dots, S_n , $DB = DB_1 \cup DB_2 \cup \dots \cup DB_n$ and DB_i locates at site S_i ($1 \leq i \leq n$). The itemset X has *local* support count of X_{sup_i} at site S_i if X_{sup_i} of the transactions contains X . The *global* support count of X is given as $X_{\text{sup}} = \sum_{i=1}^n X_{\text{sup}_i}$. An itemset X is globally supported if $X_{\text{sup}} \geq S_{\min} * \sum_{i=1}^n |DB_i|$, where S_{\min} is the user-defined minimum support. Global confidence of a rule $X \Rightarrow Y$ can be given as $\{X \cup Y\}_{\text{sup}} / X_{\text{sup}}$. The set of frequent itemsets $L_{(k)}$ consists of all k -itemsets that are globally supported. The set of locally frequent itemsets $LL_{i(k)}$ consists of all k -itemsets supported locally at site S_i . $GL_{i(k)} = L_{(k)} \cap LL_{i(k)}$ is the set of globally large k -itemsets locally supported at site S_i . The objective of distributed association rule mining is to find the itemsets $L_{(k)}$ for all $k > 1$ and the support counts for these itemsets and, based on this, generate association rules with the specified minimum support and minimum confidence. The goal of this paper is to accomplish the distributed mining process as accurately as possible without compromising the private information of local large itemsets for all the sites.

2 A Probabilistic Approach Based on Markov Chain Model

A discrete Markov chain model can be defined by the tuple $\langle S, P, \lambda \rangle$, where S corresponds to the state space, P is a matrix representing transition probabilities from one state to another, and λ is the initial probability distribution of the states in S . The fundamental property of Markov model is the dependency on the previous state. If the vector $S(t)$ denotes the probability vector for all the states at time t , then

$$\hat{S}(t) = \hat{S}(t-1) * P \quad (1)$$

If there are n states in our Markov chain, then the matrix of transition probabilities P is of size $n \times n$. Markov chains can be applied to privacy preserving data mining. In this formulation, a Markov state can correspond to a frequent k-itemset.

Let $M = \{X_1, X_2, \dots, X_m\}$ be a set of itemsets where an itemset X_i is a k-itemset. Let P_M be the itemset transition probability matrix of M subject to (1) $p_{kl} \geq 0$, (2) $\forall k (1 \leq k \leq m), \sum_{l=1}^m p_{kl} = 1$, where $p_{kl} (1 \leq k \leq m, 1 \leq l \leq m)$ is the probability with which itemset X_k transits to itemset X_l . In the distortion procedure, an itemset X_k is transited to an itemset X_l randomly with probability p_{kl} . We generate the distorted data value from a transaction by randomizing each given itemset in the transaction. The detailed procedure is given in example 1.

Example 1. Suppose a transaction t_1 in the original dataset is

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
t_1	1	0	1	1	0	1	0	1	1	0	0	0	1	0

Assume the candidate set of 2-itemsets is $\{AB, AD, BD, CD, DF, FI, HI, JK, KL, JL, MN\}$. The corresponding transition matrix P_2 is a 11×11 matrix. It is easy to find that the candidate itemsets included in t_1 are $\{AD, CD, DF, FI, HI\}$. The probability with which $\{AB\}$ is transited to $\{AB\}$ is p_{11} , $\{AB\}$ is transited to $\{AD\}$ is p_{12} , $\{AB\}$ is transited to $\{BD\}$ is p_{13} , $\{AB\}$ is transited to $\{CD\}$ is p_{14} , ..., and so on. Then based on the transition probabilities in P_2 , let's assume itemset $\{AD\}$ is transited to itemset $\{BD\}$, $\{CD\}$ to $\{DF\}$, $\{DF\}$ to $\{JK\}$, $\{FI\}$ to $\{AB\}$ and $\{HI\}$ to $\{JK\}$. Thus the transaction t_1 is distorted to $\{BD, DF, JK, AB, JK\}$.

Denoting an original itemset as X^M and the distorted itemset as X^D , the probability of correct reconstruction of itemset X_i is given by

$$R(X_i) = \sum_{j=1}^m p_{ij} * P(X^M = X_i | X^D = X_j) = \sum_{j=1}^m p_{ij}^2 * \frac{P(X^M = X_i)}{\sum_{k=1}^m p_{kj} P(X^M = X_k)}$$

where $P(X^M = X_i)$ is given by the support of X_i in the original dataset. And the probability of correct reconstruction of the set of itemsets $M = \{X_1, X_2, \dots, X_m\}$ is

$$\text{given by } R(M) = \prod_{i=1}^m \sum_{j=1}^m p_{ij}^2 * \frac{P(X^M = X_j)}{\sum_{k=1}^m p_{kj} P(X^M = X_k)}$$

Similar to [10], after computing the reconstruction probability of the set of itemsets, we can define user privacy as the following percentage:

$$P(M) = (1 - \bar{R}(M)) * 100 \quad (2)$$

where $\bar{R}(M)$ is the average reconstruction probability in a mining process. That is, when the reconstruction probability is 0, the privacy is 100%, whereas it is 0 if $\bar{R}(M) = 1$.

As mentioned earlier, the mechanism adopted in this paper for achieving privacy is to distort the user data before it is subject to the mining process. Accordingly, we measure privacy with regard to the probability with which the user's distorted items can be reconstructed.

We denote the original true set of k-itemsets by $M = \{X_1, X_2, \dots, X_m\}$ and the distorted set of k-itemsets, obtained with a distortion probability matrix P_M , as D . Let $S^M(X_1, X_2, \dots, X_m)$ be the vector of expected support of itemsets on M and $S^D(X_1, X_2, \dots, X_m)$ be the vector of support of itemsets on D . P_M is the itemset transition probability matrix of M as defined above. We have the following theorem based on the property of Markov chain.

Theorem

$$S^D(X_1, X_2, \dots, X_m) = S^M(X_1, X_2, \dots, X_m) * P_M \quad (3)$$

$$\text{And } S^M(X_1, X_2, \dots, X_m) = S^D(X_1, X_2, \dots, X_m) * P_M^{-1} \quad (4)$$

Where P_M^{-1} is the inverse matrix of P_M .

Proof. Let $C^M(X_i)$ be the support count of itemset X_i on the original dataset ($1 \leq i \leq m$). After the distortion procedure, approximately $C^M(X_1) * p_{11}$ of original X_1 will remain X_1 on the distorted dataset, $C^M(X_2) * p_{21}$ of original X_2 will transit to X_1 on the distorted dataset, $C^M(X_3) * p_{31}$ of original X_3 will transit to X_1 , ..., $C^M(X_m) * p_{m1}$ of original X_m will transit to X_1 , given $C^M(X_i)$ is large enough. Thus the overall support count of X_1 on the distorted dataset is $C^D(X_1) = \sum_{i=1}^m C^M(X_i) * p_{i1}$, and

$$S^D(X_1) = \frac{C^D(X_1)}{N} = \sum_{i=1}^m \frac{C^M(X_i)}{N} * p_{i1} = \sum_{i=1}^m S^M(X_i) * p_{i1}.$$

Similarly, the overall support of X_k ($2 \leq k \leq m$) on the distorted dataset is $S^D(X_k) = \sum_{i=1}^m S^M(X_i) * p_{ik}$. That is, $S^D(X_1, X_2, \dots, X_m) = S^M(X_1, X_2, \dots, X_m) * P_M$

Remember there are two constraints that must be satisfied when generating P_M . From these two constraints, it is easy to derive that the inverse matrix of P_M , denoted as P_M^{-1} , exists. Therefore, we have

$$S^M(X_1, X_2, \dots, X_m) = S^M(X_1, X_2, \dots, X_m) * P_M * P_M^{-1} = S^D(X_1, X_2, \dots, X_m) * P_M^{-1}$$

As[9][10] pointed out in their studies, in a probabilistic distortion approach, fundamentally we cannot expect the reconstructed support values to coincide exactly with the actual supports. This means that we may have errors in the estimated supports of frequent itemsets with the reported values being either larger or smaller than the actual supports. This kind of error is qualified as the metric of *Support Error*, ρ , which reflects the average relative error in the reconstructed support values for those itemsets that are correctly identified to be frequent. Let r_sup be the reconstructed support and a_sup be the actual support, the support error is computed over all frequent itemsets as

$$\rho = \frac{100 * \sum_f \left| \frac{r_sup_f - a_sup_f}{a_sup_f} \right|}{|f|} \quad (5)$$

Errors in support estimation can also result in errors in the identification of the frequent itemsets. It is quite likely that for an itemset slightly above $Smin$ that one of its subsets will have recovered support below $Smin$. The itemset will be discarded from the candidate set due to a key property of Apriori algorithm that if itemsets is a frequent itemset, that all of its subsets must have support larger than $Smin$. It will become especially an issue when the $Smin$ setting is such that the support of a number of itemsets lies very close to this threshold value. This kind of error is measured by the metric of *Identification Error*, which reflects the percentage error in identifying frequent itemsets and has two components: δ^+ , indicating the percentage of false positives, and δ^- indicating the percentage of false negatives. Denoting the reconstructed set of frequent itemsets with R and the correct set of frequent itemsets with F , these two metrics are computed as:

$$\delta^+ = \frac{|R - F|}{|F|} * 100 \quad \delta^- = \frac{|F - R|}{|F|} * 100 \quad (6)$$

Where $|F|$ indicates the number of frequent itemsets at k-itemset.

Hence, to reduce such errors in frequent itemset identification, we discard only those itemsets whose recovered support is smaller than a Candidate Limit, given as $Smin * (1 - \sigma)$, for candidate set generation in the algorithm we will give in Section 4. Here σ is a reduction coefficient. However, because the distortion procedure is executed in each iteration during the mining process, the error in identifying an itemset of small size will not have a similar ripple effect in terms of causing errors in identifying itemsets of longer size as presented in [10].

3 A Cryptographic Approach Based on the Pinion-Rack Encryption and Decryption Model

We put forward a database encryption model using the field of a record from a database as the basic encryption granularity and its implementation in our prior research on information security. The model is referred to as the pinion-rack encryption and decryption model (P-R Model) in terms of the pinion and rack transmission principle, as show in Fig.1.

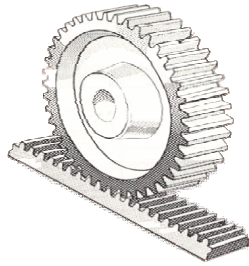


Fig. 1. Pinion and Rack Transmission Principle

There are four elements used in the encryption/decryption processes in the P-R model.

1. Encryption pinion, each tooth of the encryption pinion corresponds to an encryption algorithm and a key.
2. Decryption pinion, each tooth of the decryption pinion corresponds to a decryption algorithm and a key.
3. Plaintext rack, the database record(s) in plain text are referred to as plaintext rack, each tooth of the plaintext rack corresponds to a plain text field.
4. Ciphertext rack, the database record(s) in cipher format are referred to as ciphertext rack, each tooth of the ciphertext rack corresponds to the result of encrypting a plain text field.

During the encryption process, the encryption pinion starts from the beginning position of the plaintext rack and rotates clockwise to the end of the plaintext rack. A tooth in the plain text rack meshes with a tooth from the encryption pinion with the output of a tooth of the cipher rack, as show in Fig.2.

With this encryption process, the encrypted data in each record enjoys the following advantages: (1) Different items are encrypted with different encryption algorithms and different keys. (2) The disclosure of an encryption algorithm can not help the inference of other encryption algorithms. (3) A key can not inferred from other keys.

Similarly, in the decryption process, the decryption pinion starts from the beginning position of the ciphertext rack and rotates clockwise to the end of the ciphertext rack. A tooth in the cipher text rack meshes with a tooth from the decryption pinion with the output of a tooth of the plaintext rack, as show in Fig.3.

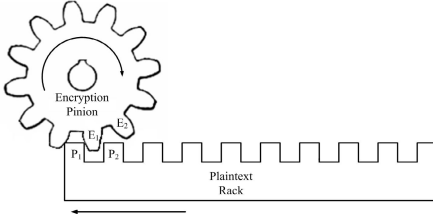


Fig. 2. Pinion-rack encryption process

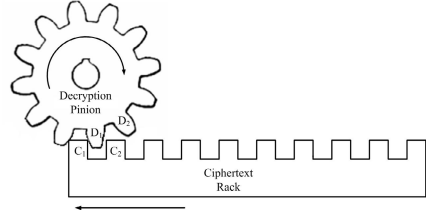


Fig. 3. Pinion-rack decryption process

In this paper, we will incorporate this encryption and decryption model with the distortion method to enhance the protection of privacy in data mining. The basic encryption granularity for our proposed algorithm, a tooth in the rack, is a k -itemset. Each local frequent itemset in the transactions is encrypted with the encryption pinion and each local frequent set of k -itemsets is also encrypted with the encryption pinion to enhance the privacy protection.

4 Privacy Preserving Mining Algorithm of Global Association Rules on Distributed Data

We will now use above cryptographic approach and probabilistic approach to construct a distributed association rule mining algorithm to preserve the privacy of individual sites. The goal of the algorithm is to find the global frequent itemsets and discover the global association rules satisfying the predefined thresholds while not, with some reasonable degree of certainty, disclosure the local frequent itemsets and the local support count of each individual site, thus protect its privacy.

The distributed association rule mining algorithm, given the global minimum support S_{min} , the global minimum confidence and the reduction coefficient σ , works as follows.

1. Let $k=1$, let the candidate set be all the single items included in the dataset.
Repeat the following steps until no itemset left in the candidate set.
{
 - (1) The common site broadcasts the transition probability matrix P_k and the encryption pinion E_k for k -itemsets included in the candidate set.
 - (2) Each site generates its local frequent k -itemsets $LL_{i(k)}$ using Apriori-like algorithm. Here the local minimum support is $S_{min} * \frac{n_i}{N}$, where n_i is the transactions on site S_i and N is the number of overall transactions on all the sites. To each transaction located in its local dataset, it finds out the frequent itemsets included in this transaction and distorts them with the distortion approach given

above. Then the distorted data is encrypted with the encryption pinion E_k and sent to the common site. The detailed distortion procedure is given in *example 1*.

- (3) The first site S_1 gets some “fake” itemsets randomly choosing from the predefined set of fake itemsets, adds them to its local frequent set of k-itemsets, encrypts its local frequent set (not the real one at this time), and then sends it to the second site. The second site adds its encrypted local frequent set, the real one without fake itemsets, to the set it gets, deletes the duplicated itemsets, and sends the set to the third site. The third and other sites do the similar work till the set is sent back to the first site. The first site removes the random fake itemsets from the set and thus gets the global candidate set. And then the global candidate set is sent to the common site for data mining.
- (4) The common site reads the encrypted and distorted dataset from all the sites, decrypts the dataset and the global candidate set using the decryption pinion D_k , and then scans the dataset to compute all the supports ($S^D(X_i)$) for each k-itemset in the global candidate set.
- (5) The common site recovers the supports on the original dataset ($S^T(X_i)$) for each k-itemset in the global candidate set using the formulae from the above theorem.
- (6) The common site discards every itemset whose support is below its candidate limit($Smin \cdot (1 - \sigma)$).
- (7) The common site saves for output only those k-itemsets $L_{(k)}$ and their supports whose recovered support is at least $Smin$.
- (8) The common site forms all possible (k+1)-itemsets such that all their k-subsets are among the remaning itemsets generated in step(6). Let these (k+1)-itemsets be the supposed new candidate set.
- (9) Let $k=k+1$.
}
2. The common site finds out all the association rules with all the saved itemsets and their supports, given the user-specified global minimum confidence. This step is straightforward and need not to be described in detail.

5 Experiments

We carry out the evaluation of the algorithm on a synthetic dataset. This dataset was generated from the IBM Almaden generator[11]. The IBM synthetic data generation program takes four parameters to create a dataset, in which, $|D|$ indicates number of transactions, $|T|$ indicates average size of the transactions, $|I|$ - average size of the maximal potentially frequent itemsets, and N - number of items. We took the parameters T10.I4.D1M.N1K resulting in a million customer tuples with each customer purchased about ten items on average. These tuples were distributed on 10

sites with a hundred thousand tuples on each site. We took a variety of $Smin$ and P_k values to evaluate the privacy and accuracy of our algorithm. [12] shows that providing high accuracy and at the same time preventing exact or partial disclosure of individual information are conflicting objectives. Therefore, in this experiment, we plan to check this point on balancing accuracy and privacy. In order to simplify the experiments, we focus the evaluation of generating frequent itemsets. The presented value of $Smin$ in this paper is 0.2%. Because the transition probability matrixes P_k are different in each iteration in our experiments, we present them in the tables. We use two different values of the reduction coefficient, namely $\sigma = 0$ and $\sigma = 10\%$ to evaluate the difference on recovery errors discussed in Section 3.5.

The results for the experiments are shown in Table 1 and Table 2. In the tables, the level indicates the length of the frequent itemset, $|F|$ indicates the actual number of frequent itemsets generated by Apriori algorithm at this level, $|F^1|$ indicates the number of frequent itemsets correctly generated by our algorithm at this level, P_k indicates the transition probability matrix simply presented in its size, ρ indicates the support error, δ^+ indicates the percentage error in identifying false positive frequent itemsets, and δ^- indicates the percentage error in false dropped frequent itemsets.

Table 1. $Smin = 0.2\%$, $\sigma = 0$

Level	$ F $	$ F^1 $	P_k	ρ	δ^+	δ^-
1	863	850	858×858	3.57	0.93	1.51
2	6780	6619	6691×6691	4.02	1.063	2.37
3	1385	1345	1359×1359	1.61	1.013	2.89
4	890	874	882×882	1.63	0.90	1.80
5	392	378	381×381	1.65	0.77	3.57
6	150	145	145×145	1.53	0	3.33
7	47	45	45×45	1.50	0	4.26
8	10	10	10×10	1.02	0	0

The user privacy value under these two conditions is, computed from formula (2), 67%, we adjust the elements of P_k in order to take the comparison under the same privacy value for these two conditions. The results in both Tables show that, even for a low minimum support of 0.2%, most of the itemsets are mined correctly from the distorted dataset. The support error is less than 5% at all levels. Note that the percentage error in identifying false positive frequent itemsets is significantly small at all levels, partially due to each iteration in the algorithm, the global candidate set is sent to the common site in an anonymous way. And because the distortion procedure is executed on the original data transactions in each iteration, the percentage error in

both false positive frequent itemsets and false negative frequent itemsets is not accumulated during the mining process.

In Table 2, the value of S_{min} is relaxed with a reduction of 10%. We can see from the results that the negative identifying error goes down significantly, at an average reduce of 0.5%, while the positive identifying error remains almost the same. The results confirm the correctness of reducing identification error through a reduction of the minimum support which we discussed in section 3.5.

Table 2. $S_{min} = 0.2\%$, $\sigma = 10\%$

Level	$ F $	$ F^1 $	P_k	ρ	δ^+	δ^-
1	863	855	859×859	3.59	0.46	0.93
2	6780	6659	6711×6711	3.97	0.77	1.78
3	1385	1367	1371×1371	2.63	0.29	1.30
4	890	879	885×885	1.79	0.67	1.24
5	392	392	393×393	2.01	0.26	0
6	150	150	150×150	1.45	0	0
7	47	47	47×47	1.48	0	0
8	10	10	10×10	0.98	0	0

Now we increase the privacy value to 86% by changing the element values of P_k and evaluate this change on the accuracy of the mining results which are shown in Table 3($\sigma = 0$).

Table 3. $S_{min} = 0.2\%$, $\sigma = 0$ with different P_k values from experiment 1

Level	$ F $	$ F^1 $		P_k	ρ	δ^+	δ^-
1	863	754	769	769×769	8.6	1.74	12.63
2	6780	5872	5941	5941×5941	8.78	1.02	13.39
3	1385	1185	1203	1203×1203	6.65	1.30	14.44
4	890	738	746	746×746	6.43	0.90	17.08
5	392	303	307	307×307	5.98	1.02	22.70
6	150	121	129	129×129	6.5	5.33	19.33
7	47	39	40	40×40	5.87	2.13	17.02
8	10	9	9	9×9	4.77	0	10

Table 3 shows that the support error ρ and the two identification errors all become much higher. For example, the false negative errors at all levels become higher than 10 with the highest at level 5. This experiment implies that the tradeoff between privacy and accuracy are very sensitive to the transition matrix. We should choose appropriate transition matrix to achieve the goal of acquiring plausible values for both privacy level and accuracy.

6 Conclusions and Discussions

In this paper, we incorporate cryptographic approach with probabilistic approach in developing data mining techniques without compromising customer privacy, and present a new algorithm for privacy preserving data mining on distributed data allocated at different sites. We focus on the task of finding frequent itemsets and extend the problem of mining association rules to distributed environments. Transactions from different sites are distorted and encrypted in order to preserve privacy. We propose a distributed privacy preserving mining algorithm and present the full process. We also evaluate the tradeoff between privacy guarantees and reconstruction accuracy and show the practicality of our approach. In our future work, we plan to complete the experiments on a real dataset and extend this approach to other applications.

References

1. Clifton, C. and Marks, D. *Security and privacy implications of data mining*. in *ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*. May 1996.
2. Estivill-Castro, V. and Brankovic, L. *Data swapping: Balancing privacy against precision in mining for logic rules*. in *Data Warehousing and Knowledge Discovery Da WaK- 99*. 1999: Springer-Verlag Lecture Notes in Computer Science.
3. Agrawal, R. *Data Mining: Crossing the Chasm*. in *the 5th International Conference on Knowledge Discovery in Databases and Data Mining*. August 1999. San Diego, California, an invited talk at SIGKDD.
4. Agrawal, R. and Srikant, R. *Privacy preserving data mining*. in *Proc. of the ACM SIGMOD Conference on Management of Data*. May 2000. Dallas, Texas.
5. Agrawal, D. and Aggarwal, C. *On the Design and Quantification of Privacy Preserving Data Mining Algorithms*. in *Proc. of 20th ACM Symp. on Principles of Database Systems (PODS)*. 2001.
6. Conway, R. and Strip, D. *Selective partial access to a database*. in *Proc. ACM Annual Conf*. 1976.
7. Breiman, L., et al. *Classification and Regression Trees*. 1984. Wadsworth, Belmont.
8. Quinlan, J. R., *Induction of decision trees*. Machine Learning, 1986. **1**: p. 81-106.
9. Evfimievski, A., et al., *Privacy Preserving Mining of Association Rules*. Information Systems, Jun 2004. **29**(4): p. 343-364.
10. Rizvi, S.J. and Haritsa, J.R. *Maintaining Data Privacy in Association Rule Mining*. in *Proc. 28th International Conf. Very Large Data Bases*. 2002.
11. Agrawal, R. and Srikant, R., *Fast Algorithms for Mining Association Rules*. June 1994, IBM Almaden Research Center: San Jose, California.
12. Adam, R. and Wortman, J. C., *Security-control methods for statistical databases*. ACM Computing Surveys, Dec. 1989. **21**(4): p. 515-556.

Multiagent Approach for the Representation of Information in a Decision Support System

Fahem Kebair and Frédéric Serin

Université du Havre, LITIS - Laboratoire d'Informatique,
de Traitement de l'Information et des Systèmes,
25 rue Philippe Lebon, 76058, Le Havre Cedex, France
{fahem.kebair, frederic.serin}@univ-lehavre.fr

Abstract. In an emergency situation, the actors need an assistance allowing them to react swiftly and efficiently. In this prospect, we present in this paper a decision support system that aims to prepare actors in a crisis situation thanks to a decision-making support. The global architecture of this system is presented in the first part. Then we focus on a part of this system which is designed to represent the information of the current situation. This part is composed of a multiagent system that is made of factual agents. Each agent carries a semantic feature and aims to represent a partial part of a situation. The agents develop thanks to their interactions by comparing their semantic features using proximity measures and according to specific ontologies.

Keywords: Decision support system, Factual agent, Indicators, Multi-agent system, Proximity measure, Semantic feature.

1 Introduction

Making a decision in a crisis situation is a complicated task. This is mainly due to the unpredictability and the rapid evolution of the environment state. Indeed, in a critic situation time and resources are limited. Our knowledge about the environment is incomplete and uncertain, verily obsolete. Consequently, it is difficult to act and to adapt to the hostile conditions of the world. This makes sense to the serious need of robust, dynamic and intelligent planning system for search-and-rescue operations to cope with the changing situation and to best save people [9]. The role of such a system is to provide an emergency planning that allows actors to react swiftly and efficiently to a crisis case.

In this context, our aim is to build a system designed to help decision-makers manage cases of crisis with an original representation of information. From the system point of view, detecting a crisis implies its representation, its characterisation and its comparison permanently with other crisis stored in scenarios base. The result of this comparison is provided to the user as the answer of the global system .

The idea began with the speech interpretation of human actors during a crisis [3], [5]. The goal was to build an *information, and communication system* (ICS)

which enables the management of emergency situations by interpreting aspects communications created by the actors. Then, a *preventive vigil system* (PVS) [1] was designed with the mean of some technologies used in the ICS modelling as: semantic features, ontologies, and agents with internal variables and behavioural automata. The PVS aims either to prevent a crisis or to deal with it with a main internal goal: detecting a crisis.

Since 2003, the architecture of the PVS was redesigned with a new specificity, that is the generic aspect; generic is used here with different meaning from [13]. A part of the global system, which is responsible of the dynamic information representation of the current situation, was applied to the game of Risk and tested thanks to a prototype implemented in Java [10]. However, we postulate that some parts of the architecture and, at a deeper level, some parts of the agents were independent of the subject used as application. Therefore, the objective at present is to connect this part to the other parts, that we present latter in this paper, and to test the whole system on various domains, as RoboCup Rescue [11] and e-learning.

We focus here on the modelling of the information representation part of the system that we intend to use it in a crisis management support system.

The paper begins with the presentation of the global system architecture. The core of the system is constituted by a multiagent system (MAS) which is structured on three multiagent layers. Then, in section 3, we explain the way we formalise the environment state and we extract information related to it, which are written in the form of semantic features. The latter constitute data that feed the system permanently and that carry information about the current situation. The semantic features are handled by factual agents and are compared the one with the other using specific ontologies [2].

Factual agents, that compose the first layer of the core, are presented thereafter in section 4. Each agent carries a semantic feature and aims to reflect a partial part of the situation. We present their structures and their behaviours inside their organisation using internal automaton and indicators.

Finally, we present a short view about the game of Risk test in which we describe the model application and the behaviour of factual agents.

2 Architecture of the Decision Support System

The role of the *decision support system* (DSS) is to provide a decision-making support to the actors in order to assist them during a crisis case. The DSS allows also managers to anticipate the occur of potential incidents thanks to a dynamic and a continuous evaluation of the current situation. Evaluation is realised by comparing the current situation with past situations stored in a scenarios base. The latter can be viewed as one part of the knowledge we have on the specific domain.

The DSS is composed of a core and three parts which are connected to it (figure 1):

- A set of user-computer interfaces and an intelligent interface allow the core to communicate with the environment. The intelligent interface controls

and manages the access to the core of the authenticated users, filters entries information and provides actors with results emitted by the system;

- An *inside query MAS* ensures the interaction between the core and world information. These information represent the knowledge the core need. The knowledge includes the scenarios, that are stored in a scenarios base, the ontologies of the domain and the proximity measures;
- An *outside query MAS* has as role to provide the core with information, that are stored in network distributed information systems.

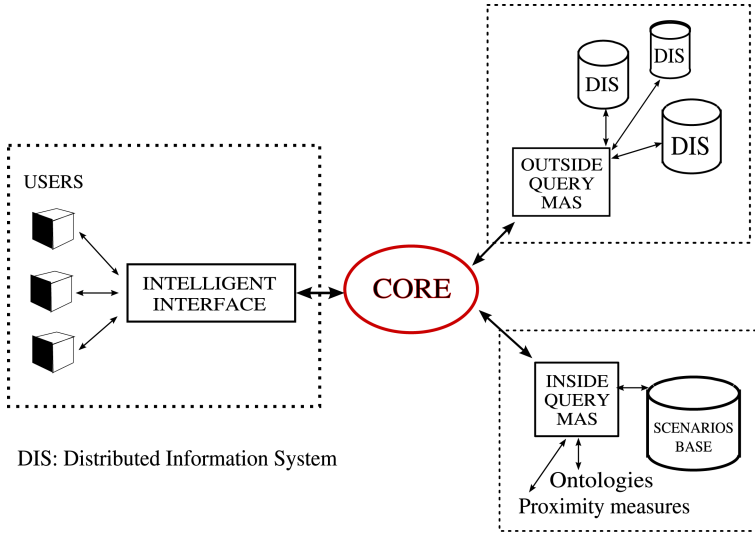


Fig. 1. General Architecture of the DSS

The core of the decision support system is made of a MAS which is structured on three layers. The latter contain specific agents that differ in their objectives and their communications way. In a first time, the system describes the semantic of the current situation thanks to data collected from the environment. Then it analyses pertinent information extracted from the scenario. Finally, it provides an evaluation of the current situation and a decision support using a dynamic and incremental case-base reasoning.

The three layers of the core are:

- The lowest layer: factual agents;
- The intermediate layer: synthesis agents;
- The highest layer: prediction agents.

Information are coming from the environment in the form of semantic features without a priori knowledge of their importance. The role of the first layer (the lowest one) is to deal with these data thanks to factual agents and let emergence

detect some subsets of all the information [7]. More precisely, the set of these agents will enable the appearance of a global behaviour thanks to their interactions and their individual operations. The system will extract thereafter from this behaviour the pertinent information that represent the salient facts of the situation.

The role of the *synthesis agents* is to deal with the agents emerged from the first layer. Synthesis agents aim to create dynamically factual agents clusters according to their evolutions. Each cluster represents an observed scenario. The set of these scenarios will be compared to past ones in order to deduce their potential consequences.

Finally, the upper layer, will build a continuous and incremental process of recollection for dynamic situations. This layer is composed of *prediction agents* and has as goal to evaluate the degree of resemblance between the current situation and its associate scenario continuously. Each prediction agent will be associated to a scenario that will bring it closer, from semantic point of view, to other scenarios for which we know already the consequences. The result of this comparison constitutes a support information that can help a manager to make a good decision.

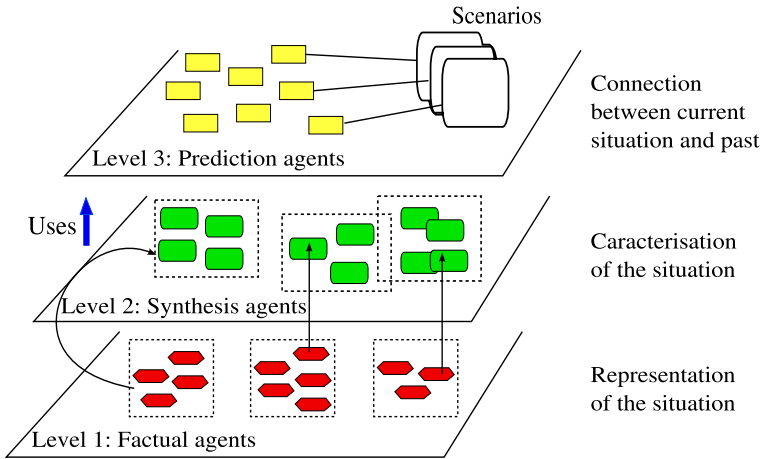


Fig. 2. Architecture of the Core

3 Environment Study and Creation of Semantic Features

3.1 Situation Formalisation

To formalise a situation means to create a formal system, in an attempt to capture the essential features of the real-world. To realise this, we model the world as a collection of objects, where each one holds some properties. The aim is to define the environment objects following the object paradigm. Therefore, we build a structural and hierarchical form in order to give a meaning to the various

relations that may exist between them. The dynamic change of these objects states and more still the interactions that could be entrenched between them will provide us a snapshot description of the environment. In our context, information are decomposed in atomic data where each one is associated to a given object.

3.2 Semantic Features

A semantic feature is an elementary piece of information coming from the environment and which represents a fact that occurred in the world. Each semantic feature is related to an object (defined in section 3.1), and allows to define all or a part of this object. A semantic feature has the following form: (key, (*qualification, value*)⁺), where key is the described object and (*qualification, value*)⁺ is a set of couples formed by: the qualification of the object and its associated value. As example of a semantic feature related to a phenomenon object: (phenomenon#1, type, fire, location, #4510, time, 9:33). The object described by this semantic feature is phenomenon#1, and has as qualifications: type, location, and time.

The modelling of semantic features makes it possible to obtain a homogeneous structure. This homogeneity is of primary importance because it allows to establish comparisons between these data. The latter are managed by factual agents, where each one carries one semantic feature and of which behaviour depends on the type of this information.

According to FIPA communicative acts [6], the agents must share the same language and vocabulary to communicate. The use of semantic features in communications process implies to define an ontology.

Inside the representation layer (the first layer of the system), agents evolve by comparing their semantic features. These comparisons allow to establish semantic distances between the agents, and are computed thanks to proximity measures.

We distinguish three types of proximities: time proximity, spatial proximity and semantic proximity. The global proximity multiplies these three proximities together. The measurement of a semantic proximity is related to ontologies. Whereas time proximity and spatial proximity are computed according to specific functions.

Proximities computation provides values on $[-1, 1]$ and is associated to a scale. The reference value in this scale is 0 that means neutral relation between the two compared semantic features. Otherwise, we can define the scale as follow: 0.4=Quiet Close, 0.7=Close, 0.9=Very Close, 1=Equal. Negative values mirrors positive ones (replacing close by different).

4 Factual Agents

4.1 Presentation and Structure of a Factual Agent

Factual agents are hybrid agents, they are both cognitive and reactive agents. They have therefore the following characteristics: reactivity, proactiveness and social ability [14]. Such an agent represents a feature with a semantic character

and has also to formulate this character feature, a behaviour [4]. This behaviour ensures the agent activity, proactiveness and communication functions.

The role of a factual agent is to manage the semantic feature that it carries inside the MAS. The agent must develop to acquire a dominating place in its organisation and consequently, to make prevail the semantic category which it represents. For this, the factual agent is designed with an implicit goal that is to gather around it as much friends as possible in order to build a cluster. In other words, the purpose of the agent is to add permanently in its acquaintances network a great number of semantically close agents. The cluster formed by these agents is recognized by the system as a scenario of the current situation and for which it can bring a potential consequence. A cluster is formed only when its agents are enough strong and consequently they are in an advanced state in their automaton. Therefore, the goal of the factual agent is to reach the action state, in which is supreme and its information may be regarded by the system as relevant.

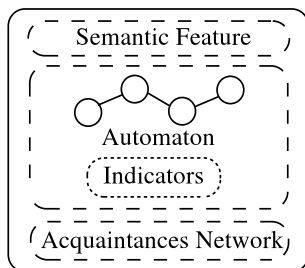


Fig. 3. Structure of a Factual Agent

An internal automaton describes the behaviour and defines the actions of the agent. Some indicators and an acquaintances network allow the automaton operation, that means they help the agent to progress inside its automaton and to execute actions in order to reach its goal. These characteristics express the proactiveness of the agent.

The acquaintances network contains the addresses of the friends agents and the enemies agents used to send messages. This network is dynamically constructed and permanently updated. Agents are friends (enemies) if their semantic proximities are strictly positive (negative).

4.2 Factual Agent Behaviour

Behavioural Automaton. The internal behaviour of a factual agent is described by a generic augmented transition network (ATN). The ATN is made of four states [3] (quoted above) linked by transitions:

- *Initialisation* state: the agent is created and enters in activities;
- *Deliberation* state: the agent searches in its acquaintances allies in order to achieve its goals;
- *Decision* state: the agent try to control its enemies to be reinforced;

- *Action* state: it is the state-goal of the factual agent, in which the latter demonstrates its strength by acting and liquidating its enemies.

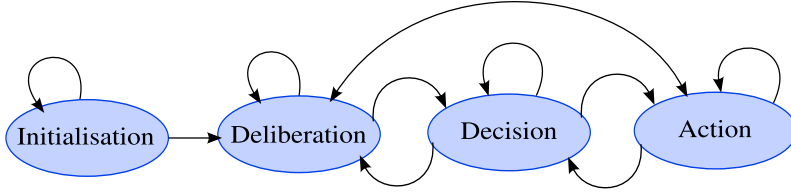


Fig. 4. Generic Automaton of a Factual Agent

ATN transitions are stamped by a set of conditions and a sequence of actions. Conditions are defined as thresholds using internal indicators. The agent must validate thus one of its outgoing current state transitions in order to pass to the next state. The actions of the agents may be an enemy aggression or a friend help. The choice of the actions to perform depend both on the type of the agent and its position in the ATN.

Factual Agent Indicators. The dynamic measurement of an agent behaviour and its state progression at a given time are given thanks to indicators. These characters are significant parameters that describe the activities variations of each agent and its structural evolution. In other words, the agent state is specified by the set of these significant characters that allow both the description of its current situation and the prediction of its future behaviour [4] (quoted above).

Factual agent has five indicators, which are pseudoPosition (PP), pseudoSpeed (PS), pseudoAcceleration (PA), satisfactory indicator (SI) and constancy indicator (CI) [8]. The “pseudo” prefix means that these indicators are not a real mathematical speed or acceleration: we chose a constant interval of time of one between two evolutions of semantic features. PP represents the current position of an agent in the agent representation space. PS evaluates the PP evolution speed and PA means the PS evolution estimation. SI is a valuation of the success of a factual agent in reaching and staying in the deliberation state. This indicator measures the satisfaction degree of the agent. Whereas, CI represents the tendency of a given factual agent to transit both from a state to a different state and from a state to the same state. This allows the stability measurement of the agent behaviour.

The compute of these indicators is according to this formulae where *valProximity* depends on the category of a given application factual agents:

$$\begin{aligned} PP_{t+1} &= valProximity \\ PS_{t+1} &= PP_{t+1} - PP_t \\ PA_{t+1} &= PS_{t+1} - PS_t \end{aligned}$$

PP, PS and PA represent thresholds that define the conditions of the ATN transitions. The definition of this conditions are specified to a given application.

As shown in the previous formulae, only PP is specific. However, PS and PA are generic and are deduced from PP. SI and CI are also independent of the studied domain and are computed according to the agent movement in its ATN.

5 Game of Risk Use Case

The first layer model has been tested on the game of Risk. We chose this game as application not only because it is well suited for crisis management but also we apprehend the elements and the actions on such an environment. Moreover we have an expert [8] (quoted above) in our team who is able to evaluate and validate results at any moment.

As result, this test proved that this model allows the dynamic information representation of the current situation thanks to factual agents organisation. Moreover we could study the behaviour and the dynamic evolution of these agents.

Risk is a strategic game which is composed of a playing board representing a map of forty-two territories that are distributed on six continents. A player wins by conquering all territories or by completing his secret mission. In turn, each player receives and places new armies and may attack adjacent territories. An attack is one or more battles fought with dice. Rules, tricks and strategies are detailed in [12].

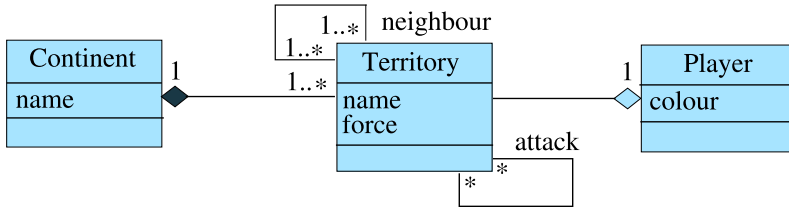


Fig. 5. Class Diagram for the Game of Risk Representation

The representation layer of the system has as role to simulate the game unwinding and to provide a semantic instantaneous description of its current state. To achieve this task, we began by identifying the different objects that define the game board (figure 5) and which are: territory, player, army and continent. Continents and territories are regarded as descriptions of a persistent situation. Whereas, armies and players are activities respectively observed (occupying a territory) and driving the actions.

From this model we distinguish two different types of semantic features: a player type and a territory type. For example (Quebec, player, green, nbArmies, 4, time, 4) is a territory semantic feature that means Quebec territory is owned by the green player and has four armies. However, (blue, nbTerritories, 4, time, 1) is a player semantic feature that signifies a blue player has four territories at step 1.

The first extracted semantic features of the initial state of the game cause the creation of factual agents. For example, a semantic feature as (red, nbTerritories, 0, time, 1) will cause the creation of red player factual agent.

During the game progression, the entry of a new semantic feature to the system may affect some agents state. A factual agent of type (Alaska, player, red, nbArmies, 3, time, 10) become (Alaska, player, red, nbArmies, -2, time, 49) with the entry of the semantic feature (Alaska, player, red, nbArmies, 1, time, 49). Alaska agent sends messages containing its semantic feature to all the other factual agents to inform them about its change. The other agents compare their own information with the received one. If an agent is interested by this message (the proximity measure between the two semantic features is not null) it updates its semantic feature accordingly. If the red player owned GB before the semantic feature (GB, player, blue, nbArmies, 5, time, 52), both red player and blue player will receive messages because of the change of the territory owner.

If we take again the preceding example (Alaska territory), Alaska agent computes its new PP (valProximity). The computation of valProximity in our case is given by: number of armies (t) - number of armies (t-1) e.g. here valProximity = 1-3 = -2. PS and PA are deduced thereafter from PP. The agent verify then the predicates of its current state outgoing transitions in order to change state. To pass from *Deliberation* state to *Decision* state for example the PS must be strictly positive. During this transition, the agent will send a *SupportMessage* to a friend and an *AgressionMessage* to an enemy.

6 Conclusion

The paper has presented a decision support system which aims to help decision-makers to analyse and evaluate a current situation. The core of the system rests on an agent-oriented multilayer architecture. We have described here the first layer which aims to provide a dynamic information representation of the current situation and its evolution in time. This part is modelled with an original information representation methodology which is based on the handle of semantic features using a factual agents organisation.

The model of the first layer was applied on the game of Risk. Results provided by this test correspond to our attempts, which consist on the dynamic representation of information. This application allowed us to track the behaviour of factual agents and to understand their parameters which are the most accurate to characterise information. Moreover, we consider that a great part of the system is generic and may be carried into other fields. Currently, we intend in a first time to connect the representation layer to the two other and to apply thereafter the whole system on more significant domains as RoboCup Rescue and e-learning.

References

1. Boukachour, H.: Système de veille préventive pour la gestion de situations d'urgence: une modélisation par organisation d'agents. Application aux risques industriels. PhD Thesis, University of le Havre, France (2002)
2. Boukachour, H., Galinho, T., Person, P., Serin, F.: Towards an architecture for the representation of dynamic situations. In Las Vegas IC-AI 659-664 (2003)

3. Cardon, A.: A multi-agent model for cooperative communications in crisis management system: the act of communication. In 7th European-Japanese Conference on Information Modeling and Knowledge-Based Simulation 111-123 (1997)
4. Cardon, A.: Modéliser et concevoir une machine pensante: approche de la conscience artificielle, Vuibert, (2004)
5. Durand, S.: Représentation des points de vue multiples dans une situation d'urgence: une modélisation par organisation d'agents. PhD Thesis, University of Le Havre, France (1999)
6. <http://www.fipa.org>, last retrieved March (2006)
7. Galinho, T., Serin, F.: Semantic Features And Factual Agents: A Model To Represent Changeable Situations A Model To Represent Changeable Situations (2003)
8. Galinho, T., Coletta, M., Person, P., Serin, F.: Dynamic Representation of Information for a Decision Support System, 8th International Conference on Enterprise Information, ICEIS2006, Paphos, Cyprus, (2006)
9. Kitano, H., Tadokoro, S.: RoboCup Rescue A Grand Challenge for Multiagent and Intelligent Systems, American Association for Artificial Intelligence (2001)
10. Person, P., Boukachour, H., Coletta, M., Galinho, T., Serin, F.: From Three MultiAgent Systems to One Decision Support System, *2nd Indian International Conference on Artificial Intelligence*, Pune, India (2005)
11. <http://sasemas.org/2005/challenge.html>, last retrieved March (2006)
<http://www.rescuesystem.org/robocuprescue>, last retrieved March (2006)
12. <http://www.thegamesjournal.com/articles/Risk.shtml>, last retrieved March (2006)
<http://www.tuxick.net/xfrisk/doc/riskfaq.html>, last retrieved March (2006)
13. Wooldridge, M., Jennings, N.R.: Pitfalls of agent-oriented development *2nd International Conference on Autonomous Agents*, pp. 385-391, Minneapolis, (1998)
14. Wooldridge, M.: *An Introduction to MultiAgent Systems*, Wiley, (2002)

Flexible Decision Making in Web Services Negotiation*

Yonglei Yao, Fangchun Yang, and Sen Su

State Key Laboratory of Networking & Switching Technology,
Beijing University of Posts & Telecommunications (BUPT) 187#, Beijing 100876, China
{yaoyl, fcyang, susen}@bupt.edu.cn

Abstract. Negotiation is a crucial stage of Web Services interaction lifecycle. By exchanging a sequence of proposals in the negotiation stage, a service provider and a consumer try to establish a formal contract, to specify agreed terms on the service, particularly terms on non-functional aspects. To react to an ever-changing environment, flexible negotiation strategies that can make adjustable rates of concession should be adopted. This paper presents such flexible strategies for Web Services negotiation. In a negotiation round, the negotiation strategies first examine the environment situations by evaluating certain factors, which include time, resources, number of counterparts and current proposals from the counterparts. For each factor, there is a corresponding function that suggests the amount of concession in terms of that factor. Then considering the importance of each service attribute, the target concession per attribute is determined. As a final step, a set of experimental tests is executed to evaluate the performance of the negotiation strategies.

Keywords: Web Services, Negotiation, Flexible Strategy, Environment.

1 Introduction

Semantic Web Services are a set of technologies that will transform the Web from a collection of static information into an infrastructure for distributed computing. Over the past several years, we have seen a growing interest in the research on Semantic Web Services. However, much of the research has concentrated on service discovery (e.g. [1]) and composition (e.g. [4]). Relatively fewer work addresses the problem of service negotiation, especially little has been done on negotiation strategies.

Negotiation is a crucial stage of the interaction lifecycle between a service consumer and a provider [5, 6]. Generally speaking, it is consisted of a sequence of proposal exchanges between the two parties, with the goal of establishing a formal contract to specify agreed terms on the service, particularly terms on non-functional aspects. Given the ability to negotiate, consumers can continuously customize their needs, and providers can tailor their offers. In particular, multiple service providers can collaborate and coordinate with each other in order to satisfy a request that they can't do alone. In short, negotiation can enable richer and more flexible interactions, and as a result, fully explore the capabilities of services.

* This work was supported by the 973 Program of China (2003CB314806), the Program for New Century Excellent Talents in University (NCET-05-0114), and the program for Changjiang Scholars and Innovative Research Team in University (PCSIRT).

When building a negotiation component, three broad areas need to be considered [7]: 1) negotiation protocol, i.e., the set of rules that govern the message exchanges between the negotiators; 2) negotiation object, which is the set of issues over which agreement must be reached. In this paper, it refers to the set of non-functional attributes of a web service and attached restrictions; 3) negotiation strategies, i.e., the set of decision-making mechanisms the negotiators employ to guide their actions in negotiation. As most of existing work on Web Service negotiation doesn't address negotiation strategies, this paper concentrates predominately on the third point, with the first two briefly defined. In particular, it engages in developing mechanisms for counter-proposal generation, when incoming proposals are not acceptable.

Operating in an environment with a high degree of uncertainty and dynamics, that is, the Internet, it is important that negotiators can act flexibly by reacting to some ever-changing environmental factors. As a preliminary work, this paper only considers factors which can influence the negotiation outcomes significantly. These include time, resources for computation and communication, number of counterparts and current proposals from counterparts. In determining the amount of concession at each negotiation round, the negotiator is guided by mathematical functions of these environmental factors and weights of different service attributes. In addition, this paper presents the results of the empirical evaluation of these negotiation strategies.

The rest of this paper is structured as follows. Section 2 presents the negotiation model. Section 3 discusses the flexible negotiation strategies and in section 4 these strategies are empirically evaluated. Section 5 summarizes related work and finally, section 6 concludes.

2 The Negotiation Model

In the Internet, a given service can be implemented by many providers. These service instances may be identical in capabilities, but have differences in non-functional properties. Consequently, a consumer can negotiate with multiple service providers concurrently, and vice versa. In addition, negotiation between the two parties involves determining a contract under multiple terms. As a result, the negotiation model for Web Services in this paper is a one-to-many, multiple issues (attributes) model, which consists of a set of bilateral, multi-attribute negotiations. The bilateral negotiation between the negotiator (a service consumer or a provider) and one of its negotiating counterparts is named as a *negotiation thread* [8]. In an individual negotiation thread, proposals and counter-proposals are generated by negotiation strategies, considering the changing environment, at each negotiation round. Note that, for an individual negotiation thread, other threads are seen as part of its environment.

2.1 The Negotiator Model

The following represents our conceptualization of a service negotiator. The model considers not only service attributes, but also attached restrictions from negotiators.

Definition 1. A service negotiator is a 4-tuple system (B, R, A, t_{\max}) , where:

1. $B = \{b_i \mid i = 1, 2, \dots, s\}$ is a set of propositions, which represents the negotiator's *profile model*. This is the background information it uses to evaluate if it can obey the counterpart's restrictions.
2. $R = \{r_j \mid j = 1, 2, \dots, h\}$ is the set of propositions that denotes the negotiator's *restriction model*, i.e., the set of restrictions that the counterpart must satisfy.
3. $A = \{(a_k, w_k, d_k, v_k) \mid k = 1, \dots, n\}$ is the negotiator's *requirement model*, which describes its preferences over non-functional attributes of the service. Where:
 - 1) a_k denotes an attribute of the service.
 - 2) w_k denotes a_k 's weight (or importance). These weights are normalized, i.e. $\sum_{1 \leq k \leq n} w_k = 1$. In general, these weights are defined by users.
 - 3) $d_k = [\min_k, \max_k]$ denotes the intervals of values for quantitative attribute a_k acceptable for the negotiator. Values for qualitative issues, on the other hand, are defined over a fully ordered domain, i.e., $d_k = \langle q_1, q_2, \dots, q_m \rangle$.
 - 4) $u_k: d_k \rightarrow [0, 1]$ is a evaluating function for issue a_k , which computes the utility of a value assigned to attribute a_k .

With these elements in place, it is possible to compute the overall utility that a negotiator can obtain if accepting a vector of values for service attributes, which is in the form of $X = \langle x_1, x_2, x_3, \dots, x_n \rangle$ (where x_k denotes a value of attribute a_k , $1 \leq k \leq n$):

$$U(X) = \sum_{k=1}^n w_k u_k(x_k) \quad (1)$$

4. t_{max} is the time deadline before which the negotiation must be terminated.

2.2 The Negotiation Thread

As mentioned above, a service negotiation consists of a set of negotiation threads, which is a bilateral bargaining process between the negotiator and one of its counterparts. In general, a negotiation thread consists of several rounds of (counter-) proposal exchanges according to the alternating-offer protocol [9]. This continues until an agreement is reached or one of the two parties terminates the negotiation without an agreement because its time deadline is reached.

Formally, a proposal from the counterpart is a 3-tuple $P = \langle B, R, X \rangle$, where B , R and $X = \langle x_k \rangle$ denotes the set of restrictions that the counterpart can obey, the set of restrictions that it requests the negotiator to satisfy and the vector of values it assigns to service attributes, respectively. The utility of p is denoted as $U(X)$.

In a negotiation round at time t , suppose a negotiator submits $P^t = \langle B^t, R^t, X^t \rangle$ and receive a proposal $P_c^t = \langle B_c^t, R_c^t, X_c^t \rangle$ from the counterpart. The negotiator first evaluates P_c^t . If both parties can satisfy each other's restrictions and $U(X_c^t) \geq U(X^t)$, P_c^t is acceptable and negotiation terminates successfully. When P_c^t is unacceptable and for some successive rounds, some restrictions requested by one party can't be satisfied by the other, the negotiator will quit the negotiation. Otherwise, the

negotiator will prepare a counter proposal, according to the strategy in section 3. As this is a preliminary work, in developing the strategies, we concentrate primarily on service attributes configuration, i.e., assigning values to all service attributes.

3 Flexible Negotiation Strategy

This section presents the strategies that a negotiator can use to prepare a proposal. For restrictions it requests the counterpart to satisfy, the negotiator will insist on the important ones, and delete those that are not important and can't be satisfied by the counterpart. For restrictions the counterpart requests, the negotiator will indicate those that it can't satisfy. For attribute values, the negotiation strategies consider factors of time, resources, and currently available counterparts to compute the amount of concession. For each factor, there is a corresponding function that suggests the value of concession in terms of that factor. Then these functions are combined, with different weights for different factors. Finally, considering importance of service attributes, the target concession per attribute is determined.

3.1 Time-Dependent Concession

As negotiation is fundamentally time-dependent, and deadlines put negotiators under pressure, it is necessary to introduce a time variable to model environment dynamics. Suppose $u_{rk} \in [0,1]$ denotes the reserved utility of attribute a_k , t is the current time and t_{max} the time deadline, and u_k^t the utility that the negotiator expects to obtain from a_k at t , then a family of time-dependent strategies can be described as:

$$u_k^t = u_{rk} + [1 - (\frac{t}{t_{max}})^{\frac{1}{\mathcal{E}}}] \times (1 - u_{rk}), \mathcal{E} > 0 \quad (2)$$

Generally, these strategies can be classified based on \mathcal{E} as follows:

1. If $0 < \mathcal{E} < 1$, the negotiator makes smaller concessions in early rounds and larger in later rounds-*conservative* strategies [11].
2. In the case of $\mathcal{E} = 1$, the negotiator makes a constant rate of concessions-*linear* strategies [11].
3. When $\mathcal{E} > 1$, the negotiator makes large concessions in the first few rounds but smaller concessions when the deadline is expiring-*conceders* [8].

Provided by the negotiator, \mathcal{E} can be used to model the eagerness of the negotiator to complete a deal [11]-the larger the \mathcal{E} , the more quickly it goes to its reservation.

Based on equation (2), if only considering time, the amount of concession on attribute a_k at time $t+1$ is computed as:

$$C_k^T = [(\frac{t+1}{t_{max}})^{\frac{1}{\mathcal{E}}} - (\frac{t}{t_{max}})^{\frac{1}{\mathcal{E}}}] \times (1 - u_{rk}), t_{max} > t+1 \quad (3)$$

3.2 Resource-Dependent Concession

Negotiation is a computational-intensive process, which is influenced by resources for computation and communication significantly. When designing negotiation strategies, the characteristics of the devices that negotiators host and underlying networks have to be considered. For example, a negotiator hosted in a mobile device should not take part in a long-time and computationally expensive negotiation, but a short one, because of the limited computational capability, while a negotiator hosted in a fixed device may follow a conservative strategy. Similarly, if the QoS of the network is low then the negotiator should concede quickly, while if the QoS is high, it may try to bargain for better deals. If other environment factors are kept unchanged, the amount of concession on attribute a_k at time $t+1$ is computed as:

$$C_k^R = (1 - r^E) \times (u_k^t - u_{rk}), \quad r \in [0, 1] \quad (4)$$

u_k^t is the utility of the value for a_k in the proposal at t , and the variable r denotes the factor of resource, which itself is parameterized with characteristics of the device and the network, e.g., CPU, memory, bandwidth, and responsive time. If resource is scarce, that is, with a low value of r , the negotiator will make a large amount of concession. Conversely, if the value of r is high, the negotiator will concede slowly.

3.3 Counterpart-Dependent Concession

As traders in real world, a negotiator's bargaining power is affected by the number of trading partners. With a large (respectively, small) number of trading partners, the negotiator will have a bargaining advantage (respectively, disadvantage) and concede slowly (respectively, quickly). In addition, difference between a negotiator's proposal and counter-proposal of its counterpart also has a significant influence on the negotiation process. The smaller (respectively larger) the difference, the higher (respectively lower) the chance of reaching an agreement is.

Suppose p_c denotes the probability of a negotiator to obtain conflict utility 0, which means that it can't complete a deal with a counterpart. In a negotiation round at time t , p_c can be formulated as [11]:

$$p_c = \frac{\prod_{i=1}^l (U^t - U_i^t)}{(U^t)^l} \quad (5)$$

Where l is the number of active counterparts at time t , U^t is the utility of the negotiator's proposal at time t and U_i^t denotes utility of a counterproposal from the i th counterpart. With other environment factors unchanged, the amount of concession on attribute a_k at time $t+1$ is computed as:

$$C_k^P = p_c \times (u_k^t - u_{rk}) \quad (6)$$

If the probability that the negotiator can't make a deal is high, it will concede more. While if this probability is low, the negotiator will concede less.

3.4 Weight-Based Target Concession Determination

At a given time, the negotiator must consider all the factors that affect the negotiation outcome, that is, she should combine them depending on the situation. Factors of time, resources for computation and communication, and behaviors of counterparts collectively determine the amount of concession on attribute a_k at time $t+1$ as:

$$C_k = W_T C_k^T + W_R C_k^R + W_P C_k^P \quad (7)$$

Where W_T , W_R and W_P denotes weight associated to factor of time, resources and counterparts, respectively. These weights can be represented by decay functions with a rate of growth α :

$$W_T = \frac{\alpha t}{\alpha t + t_{\max}}, \quad W_R = \frac{\beta t_{\max}}{\alpha t + t_{\max}}, \quad W_P = \frac{\gamma t_{\max}}{\alpha t + t_{\max}}, \quad 0 \leq t \leq t_{\max} \quad (8)$$

Where α , β and γ are positive constants and satisfy $\beta + \gamma = 1$. These functions reflect the relative importance placed on each factor according to the time left. If there is much time left, the negotiator places more importance on factors of resources and counterparts. Whereas little time left, the negotiator will adapt more to time.

As the final step, taking into account weights associated with different service attributes, the target concession on attribute a_k at time $t+1$ is computed as follows:

$$TC_k = \frac{1 - W_k}{\sum_k W_k (1 - W_k)} C_k, \quad 1 \leq k \leq n \quad (9)$$

Where W_k denotes weight (importance) of attribute a_k , and $\sum_k W_k (1 - W_k)$ is a normalized factor. Based on equation (9), the more (less) important the attribute, the less (more) concession the negotiator makes on this attribute. This coincides with intuitions of real world traders to concede more on less important terms while bargaining more on important ones.

According to equation (9), utility of attribute a_k in the counterproposal will be $u'_k - TC_k$. Based on the scoring function for a_k , u_k (see definition 1), a value x_k^{t+1} is selected for a_k , such that $u_k(x_k^{t+1}) = u'_k - TC_k$.

4 Experimental Evaluation

In this section, a set of experiments are carried out to evaluate the performance and effectiveness of the adaptive negotiation strategies. Since the flexible strategies are designed for both service consumers and providers, it suffices to evaluate the strategies from the point of view of consumers without loss of generality. For simplicity, we assume that time is discrete and is indexed by the number of negotiation rounds. In addition, since effects of eagerness, time, and trading alternatives have been deliberately evaluated in [11], here we predominately concentrate on resources, weights, and collective effects of all factors.

4.1 Effect of Resources

This experiment evaluates how resources influence the negotiation outcomes. Three tests are conducted to show the effect of different levels of resources on outcomes. In each test, one service consumer negotiates with a varying number of providers (from one to six). All providers are designed with the same initial offer, deadline, eagerness and resource. However, the three service consumers' resources are set to 0.3, 0.5, and 0.7, respectively. Experimental results are shown in figure 1:

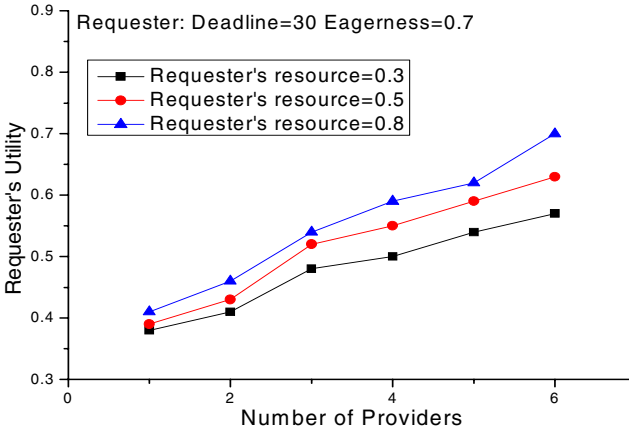


Fig. 1. Effects of Resources

Results from the tests show that service consumers with plenty of resources generally can obtain higher utility than consumers with scarce resources. Since a negotiator with scarce resources incline to make larger amount of concession, it is more likely to obtain lower utility if a consensus is reached. In addition, with the number of providers increasing, a consumer can obtain higher utility. These results are consistent with the intuitions of real world traders. When there are many negotiating alternatives and adequate resources for computation and communication, a negotiator has great bargaining "power", which makes her more aggressive and concedes slowly. If deals are made, the negotiator is likely to achieve high utility.

4.2 Effect of Weights

Six experiments are conducted to study how strategies with weight-based target concession determining process can influence the outcomes. In each experiment, a service consumer and a provider negotiate over a service with 3 attributes, namely A, B, and C. The consumer's preference over these attributes (the order of the weights associated with them) is set to <A, B, C>. However, the six providers have different preferences, namely <A, B, C>, <A, C, B>, <B, A, C>, <B, C, A>, <C, A, B> and <C, B, A>, respectively. Experimental results are shown in figure 2:

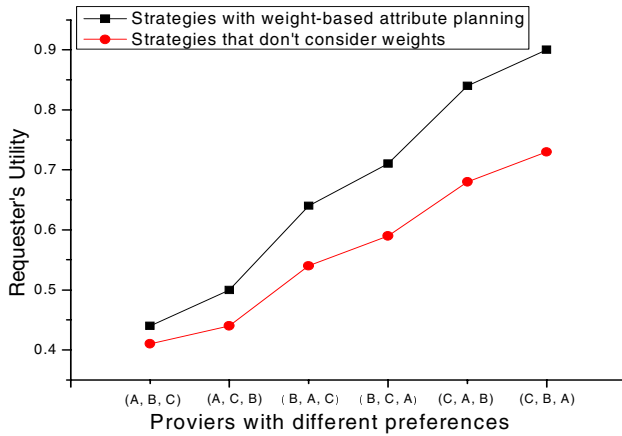


Fig. 2. Effects of Weight-Based Target Concession Determination

In all situations, the strategies with weight-based planning outperform strategies that don't consider weights of service attributes. In addition, the more difference between preferences of the two parties, the higher utility the consumer can obtain.

4.3 Collective Effect of Time, Resources and Counterparts

This section studies the collective effects of time, resources and counterparts on the outcomes. Experimental results are shown in figure 3:

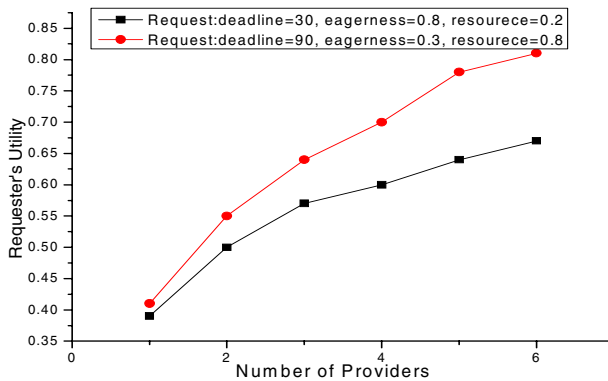


Fig. 3. Collective Effects

Though there are many combinations of these factors, due to lack of space, we only present results in extreme situations, namely situation of short deadline, high eagerness, scarce resources and situation of long deadline, low eagerness and plenty

of resources. As shown in figure 3, as deadline, resources and counterparts increasing and eagerness decreasing, the negotiator obtains more utility.

Based on the experiment results presented above, a conclusion can be drawn that our negotiation strategies provide an intuitive modeling of negotiation behaviors of real-world traders.

5 Related Work

In the area of Web services, there are increasing interests in service negotiation. However, most of existing work focuses on negotiation framework and/or negotiation protocol. Little has been done on the negotiation strategies that participants can employ to guide their actions in negotiation.

Hung et al [2] propose an XML language called WS-Negotiation for Web Services negotiation, which contains three parts: negotiation message, negotiation protocol and negotiation strategies. However, in this basic model for WS-Negotiation, the negotiation strategies are undefined.

A broker capable of offering negotiation and bargaining support to facilitate the matching of Web Services is presented in [10]. But the protocol and decision-making mechanisms for negotiation are not presented. In [3], the authors discuss their work on adapting some of the research in the multi-agent systems community to facilitate negotiation between web services. However, much of the work has concentrated on the negotiation protocol, not negotiation strategies.

WS-Agreement [12] is a recently proposed and emerging protocol for specifying contract proposals and contracts in the context of Web Services negotiation. A formal definition of the WS-Agreement is given in [13] and the schema is extended by adding tags to accommodate states for re-negotiation. In [14], WS-Agreement is extended to be semantic-enabled and suitable for accurate matching. However, the negotiation protocol is restricted to a single round “offer, accept/reject” message exchange, and they don’t develop negotiation strategies for negotiators.

In the area of Multi-Agent System, Faratin [8] and Sim [11] develop negotiation strategies which consider factors of time, counterparts, eagerness, trading opportunity, and so on. But, resources for computation and communication and weights associated with negotiation issues are not taken into account. In addition, they assume scoring functions for negotiation issues to be monotonically increasing or decreasing, which can greatly simplify the development of negotiation strategies. In contrast, our negotiator model (see definition 1) waives this assumption and makes our negotiation strategies more practical in real world applications.

6 Conclusions and Future Work

Our work makes complementary contributions to the research on Web Services negotiation. Specifically, this work proposes negotiation strategies that a service provider or consumer can employ to guide their actions in negotiation, which can react to an over-changing environment by making adjustable concessions. In more detail, taking into account of factors of time, resources and counterparts, the flexible

negotiation strategies can control concession rates and make appropriate compromises. In some situations, weight-based target concession determination can achieve better outcomes. The empirical tests show that, these approaches generally perform well and provide a more intuitive modeling of negotiation behaviors in real world.

Future work includes implementing a negotiation protocol, considering more environment factors such as the negotiation history in the design of negotiation strategies, improving the flexible negotiation strategies with learning capabilities, and developing negotiation strategies operating in the context of service selection and composition.

References

- [1] Z. Shen, J. Su, Web service discovery based on behavior signatures, Proc. 2005 IEEE International Conference on Services Computing, Volume 1, pp 279 - 286, 11-15 July 2005
- [2] Patrick C. K. Hung, Haifei Li, Jun-Jang Jeng, WS-Negotiation: an overview of research issues, Proc. HICSS 2004, Jan. 5-8, 2004
- [3] S. Paurobally, V. Tamma, and M. Wooldridge, Cooperation and Agreement between Semantic Web Services, W3C Workshop on Frameworks for Semantics in Web Services, Innsbruck, Austria, June 2005.
- [4] Z. Maamar, S.K. Mostefaoui and H. Yahyaoui, Toward an Agent-Based and Context-Oriented Approach for Web Services Composition, Knowledge and Data Engineering, IEEE Transactions on, Volume 17, Issue 5, May 2005, pp 686 - 697
- [5] C. Preist, A Conceptual Architecture for Semantic Web Services, In proceeding of International Semantic Web Conference, Hiroshima, Japan, 8-11 November 2004
- [6] M. Burstein, C. Bussler, T. Finin, M.N. Huhns, M. Paolucci, A.P. Sheth, S. Williams and M. Zaremba, A semantic Web services architecture, Internet Computing, Volume 9, Issue 5, pp 72-81, Sept.-Oct. 2005
- [7] A. R. Lomuscio, M. Wooldridge and N. R. Jennings, A classification scheme for negotiation in electronic commerce, Int. Journal of Group Decision and Negotiation, Volume 12, issue 1, pp 31-56, 2003
- [8] P. Faratin, C. Sierra, and N.R Jennings, Negotiation decision functions for autonomous agents, Robotics and Autonomous Systems, 24(3-4):159-182, 1998.
- [9] A. Rubinstein, Perfect equilibrium in a bargaining model, Econometrica 50(1), 1982
- [10] Tung Bui and A. Gachet, Web Services for Negotiation and Bargaining in Electronic Markets: Design Requirements and Implementation Framework, Proceedings of the 38th Annual Hawaii International Conference on System Sciences, pp 38 - 38, 03-06 Jan, 2005
- [11] K.M. Sim, S.Y. Wang, Flexible negotiation agent with relaxed decision rules, IEEE Transactions on Systems, Man and Cybernetics, Part B, Volume 34, Issue 3, Page(s):1602 - 1608, June 2004
- [12] A. Andrieux, C. Czajkowski, A. Dan, K. Keahey, H. Ludwig, J. Pruyne, J. Rofrano, S. Tuecke, M. Xu, Web Services Agreement Specification (WS-Agreement), June 29 2005
- [13] M. Aiello, G. Frankova, and D. Malfatti, What's in an Agreement? An Analysis and an Extension of WS-Agreement, Proc. 3rd ICSOC, 2005
- [14] N. Oldham, K. Verma, A. Sheth, F. Hakimpour, Semantic WS-Agreement Partner Selection, Proc. WWW 2006, Edinburgh Scotland, May 23-26, 2006

On a Unified Framework for Sampling With and Without Replacement in Decision Tree Ensembles

J.M. Martínez-Otzeta, B. Sierra, E. Lazkano, and E. Jauregi

Department of Computer Science and Artificial Intelligence
University of the Basque Country

P. Manuel Lardizabal 1, 20018 Donostia-San Sebastián

Basque Country, Spain

`ccbmaotj@si.ehu.es`

<http://www.sc.ehu.es/ccwrobot>

Abstract. Classifier ensembles is an active area of research within the machine learning community. One of the most successful techniques is *bagging*, where an algorithm (typically a decision tree inducer) is applied over several different training sets, obtained applying sampling with replacement to the original database. In this paper we define a framework where sampling with and without replacement can be viewed as the extreme cases of a more general process, and analyze the performance of the extension of *bagging* to such framework.

Keywords: Ensemble Methods, Decision Trees.

1 Introduction

One of the most active areas of research in the machine learning community is the study of *classifier ensembles*.

Combining the predictions of a set of component classifiers has been shown to yield accuracy higher than the most accurate component on a long variety of supervised classification problems. To achieve this goal, various strategies of combining these classifiers in different ways are possible [Xu et al., 1992] [Lu, 1996] [Dietterich, 1997] [Bauer and Kohavi, 1999] [Sierra et al., 2001]. Good introductions to the area can be found in [Gama, 2000] and [Gunes et al., 2003]. For a comprehensive work on the issue see [Kuncheva, 2004].

The combination, mixture, or ensemble of classification models can be performed mainly by means of two approaches:

- Concurrent execution of some paradigms with a posterior combination of the individual decision each model has given to the case to classify [Wolpert, 1992]; the combination can be done by voting or by means of more complex approaches [Ho et al., 1994].
- Hybrid approaches, in which two or more different classification systems are implemented together in one classifier [Kohavi, 1996].

When implementing a model belonging to the first approach, a necessary condition is that the ensemble classifiers are *diverse*. One of the ways to achieve this consists of using several base classifiers, apply them to the database, and then combine their predictions in a single one. But even with a unique base classifier, is still possible to build an ensemble, applying it to different training sets in order to generate several different models.

One of the ways to get several training sets from a given dataset is *bootstrap* sampling, where a sampling with replacement is made, obtaining samples with the same cardinality than the original dataset, but with different composition (some instances from the original set may be missing, while others may appear more than once).

This is the method that *bagging* [Breiman, 1996] uses to obtain several training databases from a unique dataset. In this work we present a sampling method that make appear sampling with and without generalization as the two extreme cases of a more general continuous process. Then, it is possible to analyze the performance of *bagging* or any other algorithm that makes use of sampling with or without replacement in the continuum that spans between the two extremes.

Typically, the base classifier in a given implementation of *bagging* uses to be a decision tree, due to the fact that small changes in the training data use to lead to proportionally big changes in the built tree.

A *decision tree* consists of nodes and branches to partition a set of samples into a set of covering decision rules. In each node, a single test or decision is made to obtain a partition. In each node, the main task is to select an attribute that makes the best partition between the classes of the samples in the training set. In our experiments, we use the well-known decision tree induction algorithm, C4.5 [Quinlan, 1993].

The rest of the paper is organized as follows. Section 2 presents the proposed framework, with a brief description of the bagging algorithm. In section 3 the experimental setup in which the experiments were carried out is described. The obtained results are shown in section 4 and section 5 is devoted to conclusions and further work.

2 Unified Framework

In this section we will define a sampling process, of which sampling with replacement, and without replacement are the two extreme cases, existing a continuous range of intermediate possibilities.

To define the general case, first of all let us take a glance to sampling with and without replacement:

- Sampling with replacement: an instance is sampled according to some probability function, it is recorded, and then returned to the original database
- Sampling without replacement: an instance is sampled according to some probability function, it is recorded, and then discarded

Let us define now the following sampling method:

- Sampling with probability p of replacement: an instance is sampled according to some probability function, it is recorded, and then, *with a probability p* , returned to the original database

It is clear than the last definition is more general that the other two, and includes them.

The differences among the three processes are depicted in Figure 1.

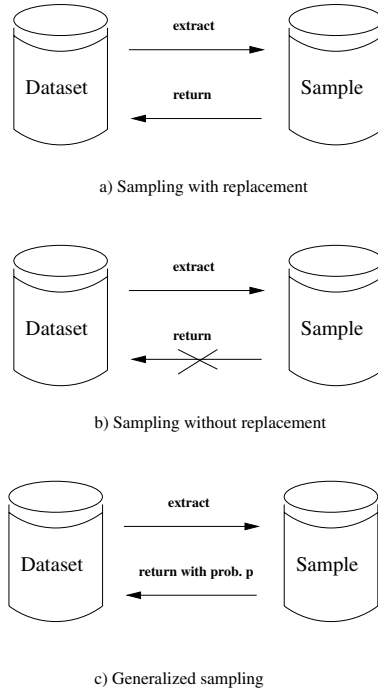


Fig. 1. Graphical representation of the three different sampling methods

As we have already noted, sampling with replacement is one of the extreme cases of the above definition: when p is 1, so every instance sampled is returned to the database. Sampling without replacement is the opposite case, where p is 0, so a sampled instance is discarded and never returns to the database.

Some questions arise: is it possible to apply this method to some problem where sampling with replacement is used to overcome the limitations of sampling without replacement? Will be the best results obtained when p is 0 or 1, or maybe in some intermediate point?

We have tested this generalization in one well-known algorithm: *bagging*, where a sampling with replacement is made.

2.1 Bagging

Bagging (*bootstrap aggregating*) is a method that adds up the predictions of several classifiers by means of voting. These classifiers are built from several training sets obtained from a unique database through sampling with replacement.

Leo Breiman described this technique in the early 90's and it has been widely used to improve the results of single classifiers, specially decision trees.

Each individual model is created from a instance set with the same number of elements than the original one, but obtained through sampling with replacement. Therefore, if in the original database there are m elements, every element has a probability $1 - (1 - 1/m)^m$ of being selected at least once in the m times sampling is performed. The limit of this expression for big values of m is $1 - 1/e$, which yields the value of 63.2. Therefore, in average, only a 63.2% of the original cases will be present in the new set, appearing some of them several times.

Bagging Algorithm

- Initialize parameters
 - Initialize the set of classifiers $D = \emptyset$
 - Initialize N , the number of classifiers
- For $n = 1, \dots, N$:
 - Extract a sample B_n through sampling with replacement from the original database
 - Built a classifier D_n taking B_n as training set
 - Add the classifier obtained in the previous step to the set of classifiers:
 $D = D \cup D_n$
- Return D

It is straightforward to apply the previously introduced approach to *bagging*. The only modification in the algorithm consists in replacing the standard sampling procedure by the generalization above described.

3 Experimental Setup

In order to evaluate the performance of the proposed sampling procedure, we have carried out an experiment over a high number of the well-known UCI repository databases [Newman et al., 1998]. To do so, we have selected all the databases of medium size (between 100 and 1000 instances) among those converted to the $\mathcal{MLC}++$ [Kohavi et al., 1997] format, and located in this public repository: [<http://www.sgi.com/tech/mlc/db/>] This amounts to 59 databases,

Table 1. Characteristics of the 41 databases used in this experiment

<i>Database</i>	<i>#Instances</i>	<i>#Attributes</i>	<i>#Classes</i>
Anneal	798	38	6
Audiology	226	69	24
Australian	690	14	2
Auto	205	26	7
Balance-Scale	625	4	3
Banding	238	29	2
Breast	699	10	2
Breast-cancer	286	9	2
Cars	392	8	3
Cars1	392	7	3
Cleve	303	14	2
Corral	129	6	2
Crx	690	15	2
Diabetes	768	8	2
Echocardiogram	132	7	2
German	1000	20	2
Glass	214	9	7
Glass2	163	9	2
Hayes-Roth	162	4	3
Heart	270	13	2
Hepatitis	155	19	2
Horse-colic	368	28	2
Hungarian	294	13	2
Ionosphere	351	34	2
Iris	150	4	3
Liver-disorder	345	6	2
Lymphography	148	19	4
Monk1	432	6	2
Monk2	432	6	2
Monk3	432	6	2
Pima	768	8	2
Primary-org	339	17	22
Solar	323	11	6
Sonar	208	59	2
ThreeOf9	512	9	2
Tic-tac-toe	958	9	2
Tokyo1	963	44	2
Vehicle	846	18	4
Vote	435	16	2
Wine	178	13	3
Zoo	101	16	7

from which we have selected one of each family of problems. For example, we have chosen *monk1* and not *monk1-cross*, *monk1-full* or *monk1-org*. After this final selection, we were left with the 41 databases shown in Table 1.

```

begin Generalized sampling testing
Input: 41 databases from UCI repository
For every database in the input
  For every  $p$  in the range 0..1 in steps 0.00625
    For every fold in a 10-fold cross validation
      Construct 10 training sets sampling
        according to parameter  $p$ 
      Induce models from those sets
      Present the test set to every model
      Make a voting
      Return the ensemble prediction and accuracy
    end For
  end For
end For
end Generalized sampling testing

```

Fig. 2. Description of the testing process of the generalized sampling algorithm

The sampling generalization described in the previous section makes use of a parameter p that is continuous. In the experiments carried out we have tested the performance of every value of p between 0 and 1 in steps of 0.00625 width. this amounts to a total of 161 discrete values.

For every value of p a 10-fold crossvalidation has been carried out.

In Figure 2 is depicted the algorithm used for the evaluation.

4 Experimental Results

In this section we present the experimental results obtained from a experiment following the methodology described in the previous section.

We were interested in analyze the performance of the modification of *bagging* when sampling with replacement was changed by our approach, and for what values of p better results are obtained. Therefore, to analyze the data of all the databases, we have normalized the performances, taking as unit the case where p is zero. This case is equivalent to sampling without replacement, so it is clear that every set obtained in this way from a given dataset will be equivalent to the original one. This case corresponds to apply the base classifier (in our case C4.5) without any modification at all. This will be our reference when comparing performances with other p values.

For example if, over the dataset A, with $p = 0$ we obtain an accuracy of 50%, and with $p = 0.6$ the performance is 53%, the normalized values would be 1 and 1.06 respectively. In other words, the accuracy in the second case is a six per cent better than in the first one. This normalization will permit us analyze which values of p yield better accuracy with respect to the base classifier.

Standard *bagging* is the case when p takes the value 1. The obtained databases are diverse and this is one of the causes of the expected better performance. But,

is this gain in performance uniform over all the interval? Our results show that it is not the case, and that beyond $p = 0.5$ there are no noticeable gains, being the most important shifts around $p = 0.15$ and $p = 0.25$. This means that small diversity between samplings could lead to similar results than the big diversity that *bagging* produces.

After normalization, each database would have associated a performance (typically between 0.9 and 1.1) to every value of p ; this performance is the result of the 10-fold crossvalidation as explained in the previous section. After applying linear regression, we obtained the results shown below.

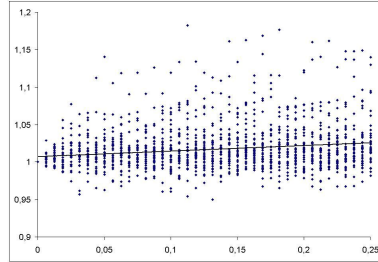


Fig. 3. Regression line for values of p between 0 and 0.25: $y = 0.0763x + 1.0068$

In every figure, the X axe is p , while Y is the normalized performance. In Figures 3, 4, 5 and 6 are shown the results in the intervals $(0, 0.25)$, $(0.25, 0.5)$, $(0.5, 0.75)$ and $(0.75, 1)$, respectively. In every interval, the normalization has been carried out with respect to the lower limit of the interval. This has been made to make clear the gains in that interval.

Observing the slope of the regression line, we note that the bigger gains are in the first interval. In the second one, there are some gains, but not at all in the same amount than in the first one. In the two last intervals the gains are very small, if any.

Let us note too that this means that the cloud of points in Figure 3 is skewed towards bigger performance values than the clouds depicted in Figures 4, 5 and 6. The extreme lower values in Figure 3 are around 0.95, while in the other intervals appear some values below that limit. This means the chances of an important drop in performance are much smaller than the opposite.

With respect to Figure 4, where some perceptible gains are still achieved, it is observed that performances below 0.92 are extremely rare, while in Figure 5 appear a higher amount of them. In Figure 6, apart from a unique extreme case below 0.90, the frequency of appearance of performances below 0.92 is very rare too. More detailed analysis is needed to distinguish true patterns behind this data from statistical fluctuations.

From these data it looks as if with little diversity it is possible to achieve the same results than with *bagging*. In Figure 7 it is drawn the result of a polynomial regression of sixth degree. It shows that values close to the maximum are around $p = 0.15$.

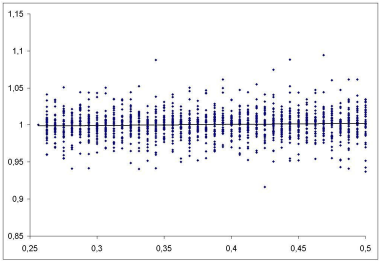


Fig. 4. Regression line for values of p between 0.25 and 0.50: $y = 0.0141x + 0.9951$

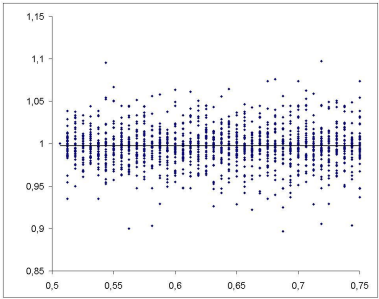


Fig. 5. Regression line for values of p between 0.50 and 0.75: $y = 0.0008x + 0.9969$

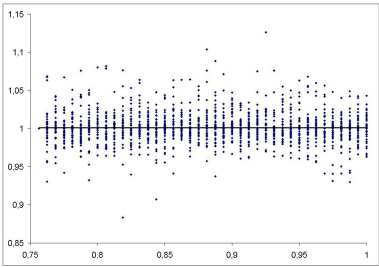


Fig. 6. Regression line for values of p between 0.75 and 1: $y = -0.0004x + 1.0017$

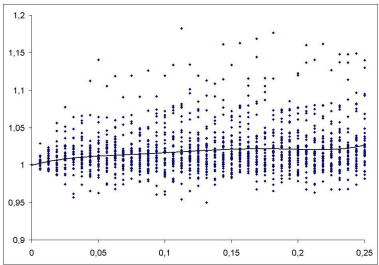


Fig. 7. Sixth grade polynomial regression for values of p between 0 and 0.25

5 Conclusions and Further Work

In this paper we have defined a generalization of sampling that includes sampling with and without replacement as extreme cases. This sampling has been applied to the bagging algorithm, in order to analyze its behavior. The results suggests that ensembles with less diversity than those obtained applying bagging could achieve similar performances.

The analysis carried out in previous sections has been made over the accumulated data of all the 41 databases, so another line of research could consist of detailed analysis of performance over any given database. In this way, it could be possible a characterization of databases for which improvements in the interval $(0, 0.25)$ are more noticeable and, in the other hand, databases for which improvements are achieved in intervals different than $(0, 0.25)$. Let us note that the above results have been obtained putting together the 41 databases, so it is expected that some databases will behave different than the main trend; in some cases they will be against the main behavior, and in others their results will be in the same line, but much more marked.

As further work, a better analysis of the interval $(0, 0.25)$, where the most dramatic changes occur, would be of interest.

A study of the value of similarity measures when applied over the ensembles obtained with different p values would be desirable too, along with theoretical work.

Acknowledgments

This work has been supported by the Ministerio de Ciencia y Tecnología under grant TSI2005-00390 and by the Gipuzkoako Foru Aldundia OF-838/2004.

References

- [Bauer and Kohavi, 1999] Bauer, E. and Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting and variants. *Machine Learning*, 36(1-2):105–142.
- [Breiman, 1996] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.
- [Dietterich, 1997] Dietterich, T. G. (1997). Machine learning research: four current directions. *AI Magazine*, 18(4):97–136.
- [Gama, 2000] Gama, J. (2000). *Combining Classification Algorithms*. Phd Thesis. University of Porto.
- [Gunes et al., 2003] Gunes, V., Ménard, M., and Loonis, P. (2003). Combination, co-operation and selection of classifiers: A state of the art. *International Journal of Pattern Recognition*, 17(8):1303–1324.
- [Ho et al., 1994] Ho, T. K., Hull, J. J., and Sridhari, S. N. (1994). Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:66–75.
- [Kohavi, 1996] Kohavi, R. (1996). Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*.

- [Kohavi et al., 1997] Kohavi, R., Sommerfield, D., and Dougherty, J. (1997). Data mining using $\mathcal{MLC}++$, a machine learning library in $C++$. *International Journal of Artificial Intelligence Tools*, 6(4):537–566.
- [Kuncheva, 2004] Kuncheva, L. I. (2004). *Combining pattern classifiers: methods and algorithms*. Wiley-Interscience, Hoboken, New Jersey.
- [Lu, 1996] Lu, Y. (1996). Knowledge integration in a multiple classifier system. *Applied Intelligence*, 6(2):75–86.
- [Newman et al., 1998] Newman, D., Hettich, S., Blake, C., and Merz, C. (1998). UCI repository of machine learning databases.
- [Quinlan, 1993] Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.
- [Sierra et al., 2001] Sierra, B., Serrano, N., Larrañaga, P., Plasencia, E. J., Inza, I., Jiménez, J. J., Revuelta, P., and Mora, M. L. (2001). Using bayesian networks in the construction of a bi-level multi-classifier. *Artificial Intelligence in Medicine*, 22:233–248.
- [Wolpert, 1992] Wolpert, D. (1992). Stacked generalization. *Neural Networks*, 5:241–259.
- [Xu et al., 1992] Xu, L., Kryzak, A., and Suen, C. Y. (1992). Methods for combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on SMC*, 22:418–435.

Spatio-temporal Proximities for Multimedia Document Adaptation

Sébastien Laborie

INRIA Rhône-Alpes – 655, Avenue de l'Europe – 38334 Saint Ismier – France
sebastien.laborie@inrialpes.fr

Abstract. The multiplication of execution contexts for multimedia documents requires the adaptation of the document specification to the particularities of the contexts. In this paper, we propose to apply a semantic approach for multimedia document adaptation to the spatio-temporal dimension of documents. To guarantee that the adapted document is close to the initial one respecting adaptation constraints, we define proximities for adapting static documents (*i.e.*, documents without animations) and animated documents. Moreover, we show that these proximities can be refined according to multimedia object properties (*e.g.*, images, videos...). The approach is illustrated by an example.

Keywords: Knowledge representation and reasoning, semantic adaptation.

1 Introduction

With the proliferation of heterogeneous devices (*e.g.*, desktop computers, personal digital assistants, mobile phones, setup boxes...), multimedia documents must be adapted under various constraints (*e.g.*, small screens, low bandwidth...). [1] proposed a semantic approach for multimedia document adaptation. This approach does not deal with the semantics of the document content, but with that of its composition. It mainly consists of specifying semantic relations between multimedia objects, and then providing adapted documents, which are close to the initial one respecting adaptation constraints, by transforming these relations if necessary. This framework has been applied to the temporal [1] and spatial [2] dimensions of multimedia documents.

This paper goes one step further by combining the temporal and spatial dimensions of documents. We aim at adapting multimedia documents along their spatio-temporal dimension. The key idea of the paper is not only to apply the adaptation approach to an additional dimension, but mainly to define new metrics in this context denoting proximities between documents. Thus, we define spatio-temporal proximities for static documents (*i.e.*, without animations) and animated documents. Moreover, we refine these proximities according to multimedia object properties (*e.g.*, images, videos...).

The organization of the paper will be as follow. Section 2 is devoted to multimedia document specification. We present a multimedia document example and show its spatio-temporal dimension. Section 3 introduces the multimedia document adaptation problem and motivates the semantically grounded adaptation approach. Section 4 presents a spatio-temporal representation on which we define, in the following sections, proximities. Section 5 presents a semantic framework for multimedia document adaptation and its extension to handle the spatio-temporal representation. We define metrics,

denoting proximities, for static and animated documents. Section 6 refines these metrics according to multimedia object properties.

2 Multimedia Document Specification

Multimedia documents are defined by their temporal, spatial, logical and interactive dimensions [3]. This paper primarily focuses on the adaptation of multimedia documents along their spatio-temporal dimension. The organization of such a document over time and space is presented in Fig. 1. The example is a multimedia presentation of a boxing match "Jake La Motta vs. Sugar Ray Robinson" which took place at the Chicago Stadium on February 14th 1951. The presentation is composed of various multimedia objects: a video of the match, the speech of the speaker, the name of one of the boxers and his different results. Each of these objects is organized over time. For instance, the video of the match starts at second 0 and ends at second 44. The name of one of the boxers appears at second 3 and disappears at second 8...

Visible objects are organized over space. In this example, they are represented by a rectangle defined with an origin point (in this case it is the upper left point), a height and a width. For instance, the object *Name* has 226 pixels width, 30 pixels height and an origin point situated at (90, 250). The object *Results* has 100 pixels width, 110 pixels height and an origin point situated at (280, 10)...

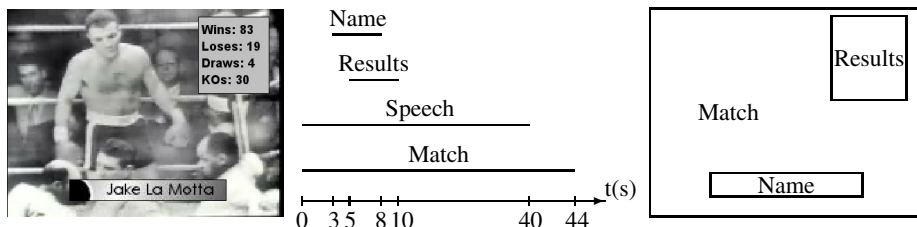


Fig. 1. Boxing match (left), temporal (middle) and spatial (right) dimensions

Such descriptions are exact and quantitative since they define exactly the presentation of each multimedia object. This information is sufficient for playing the document: only one possible execution of the document corresponds to each exact quantitative representation.

The dimensions of multimedia documents are only seldom specified in this exact way because it is more convenient for the author to leave the interpretation of the specification to the machine as long as the author intention is clearly expressed and satisfied. Expressing the purely qualitative relations between multimedia objects leads to non-precise specifications, *e.g.*, the name is presented *during* the match or the results are presented *above* the name...

There are several languages for specifying multimedia documents with different ways of describing the temporal and spatial dimensions. For example, SMIL [4] uses both qualitative and quantitative relations, while Madeus [5] uses qualitative relations.

3 Multimedia Document Adaptation

A multimedia document may be played on different devices with different capabilities: phones, PDAs, desktop computers, setup boxes, etc. These introduce different constraints on the presentation itself. For instance, display limitations (*e.g.*, mobile phones) can prevent overlapped visible objects to be displayed at a time for visibility reasons. This constraint had already been identified in [6]. However, they focused their works on multimedia document authoring instead of multimedia document adaptation. Other constraints may also be introduced by user preferences, content protection or terminal capabilities [7]. The set of constraints imposed by a client is called a profile.

To satisfy these constraints, multimedia documents must be adapted, *i.e.*, transformed into documents compatible with the target contexts before being played. Several kinds of adaptation are possible, such as local adaptation (*i.e.*, adaptation of media objects individually) and global adaptation (*i.e.*, adaptation of the document structure). This paper will focus on the latter.

The adaptation is then usually performed by a program transforming the document, *e.g.*, [8]. It could be implicit, if we have alternative solutions. Nevertheless, it is necessary to know in advance the different target profiles. Adaptation could also be explicit, *i.e.*, using the semantic of the document. Qualitative specifications are central to this process as they enable more efficient adaptation by providing more flexibility. In the next section, we present a qualitative spatio-temporal representation.

4 A Qualitative Spatio-temporal Representation

We consider a set of spatio-temporal relations as a set of pairs $\langle r_T, r_S \rangle$ where r_T is a temporal relation from the set \mathcal{T} ($r_T \neq \emptyset$) and r_S is a spatial relation from the set \mathcal{S} (r_S is possibly empty, *e.g.*, in Fig. 1 the Speech has no spatial relation with the other multimedia objects). In this paper, \mathcal{T} is the set of Allen relations [9]:

relation (r): $x \ r \ y$	x / y	inverse: $y \ r^{-1} \ x$
before (b)	— —	(bi) after
meets (m)	— —	(mi) met-by
during (d)	==	(di) contains
overlaps (o)	— =	(oi) overlapped-by
starts (s)	==	(si) started-by
finishes (f)	— =	(fi) finished-by
equals (e)	==	(e)

\mathcal{S} is the set of RCC8 relations [10] presented in Fig. 2.

5 Spatio-temporal Multimedia Document Adaptation

In section 5.1, the general adaptation approach of [1] is presented. Section 5.2 defines spatio-temporal proximities between documents, and used these proximities for multimedia document adaptation. Section 5.3 considers animated documents.

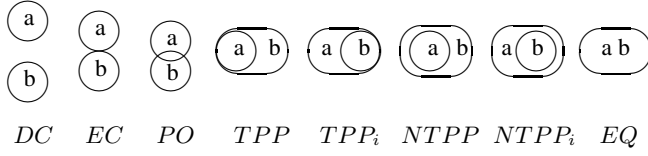


Fig. 2. The RCC8 relations

5.1 The General Approach

In [1], a semantic approach for multimedia document adaptation is defined. This approach interprets each document as the set of its potential executions (*i.e.*, related to the initial document) and a profile as the set of possible executions (*i.e.*, respecting adaptation constraints). In this context, “adapting” amounts to find the set of potential executions that are possible. When none is possible, the goal of adaptation is to find executions as close as possible to potential executions that satisfy the profile.

We consider both the multimedia document specifications and the profiles as a set of relations holding between multimedia objects. The potential and possible executions are ideally represented by relation graphs.

Definition 1 (Relation graph). A multimedia document specification $s = \langle O, C \rangle$ relative to a set of executions, with O the set of multimedia objects and C the set of constraints between the elements of O , can be represented as a complete directed labeled graph $g_s = \langle N, E, \lambda \rangle$, called a relation graph. The elements of O are in bijection with those of N and $\lambda : E \rightarrow 2^{\mathcal{R}}$ is a total function from the arcs to the set of relations (here $\mathcal{R} = \mathcal{T} \times \mathcal{S}$) such that for each $x \ r \ y \in C$, $\lambda(\langle x, y \rangle) \subseteq r$.

Fig. 3 presents two relation graphs. Each node corresponds to a multimedia object and each arc is labeled by a set of spatio-temporal relations (inverse relations are not noted). The potential executions (left) include, in particular, the execution of Fig.1 (*i.e.*, the initial document). The possible executions (right) correspond to the following profile: overlapping visible objects are impossible at a time. It may occur that some potential relations are not possible.

In this context, adapting consists of finding a set of relation graphs corresponding to possible executions (*i.e.*, respecting adaptation constraints) at a minimal distance from

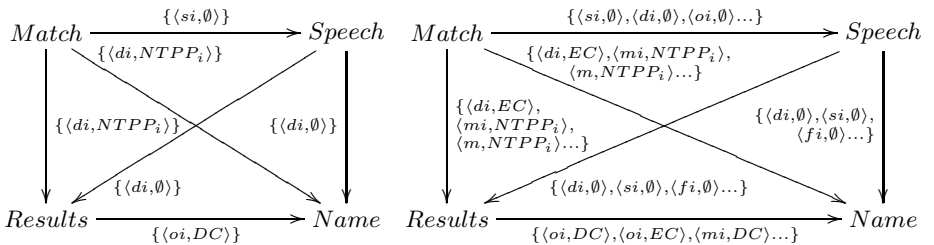


Fig. 3. Potential (left) and possible (right) executions

the relation graph of potential executions (*i.e.*, the initial document specification). This set of relation graphs is thus called adapted relation graphs solutions.

We consider that the proximity between two relation graphs is based on the variation of the relations labeling the edges of the graphs. It takes advantage of the conceptual neighborhood structure between relations denoting their proximity. This structure is usually represented by a conceptual neighborhood graph. Two conceptual neighborhood graphs are presented in section 5.2 (Fig. 4).

Thus, a conceptual neighborhood distance δ between two relations is computed from their shortest path distance in the conceptual neighborhood graph. Moreover, a distance d between relation graphs is obtained by summing up all the conceptual neighborhood distances between relationships used in both relation graphs (Def. 2).

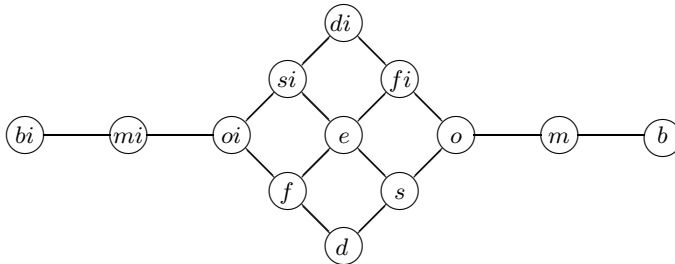
Definition 2 (Conceptual neighborhood distance between relation graphs).

$$d(\lambda, \lambda') = \sum_{n, n' \in N} \text{Min}_{r \in \lambda(\langle n, n' \rangle), r' \in \lambda'(\langle n, n' \rangle)} \delta(r, r')$$

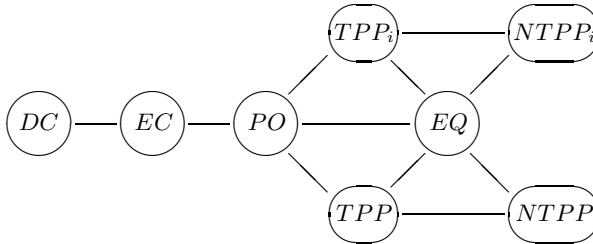
This approach has been fully defined for the temporal and spatial cases. Our goal is to define a similar procedure applying to the spatio-temporal dimension of multimedia documents. Thus, defining spatio-temporal proximities.

5.2 Spatio-temporal Proximities

To find a relation graph respecting adaptation constraints as close as possible to the relation graph corresponding to potential executions, proximities between spatio-temporal relations are defined. Proximities between Allen and RCC8 relations (*resp.*, Fig. 4(a)



(a) Conceptual neighborhood graph of Allen relations.



(b) Conceptual neighborhood graph of RCC8 relations.

Fig. 4. Conceptual neighborhood Graphs

and Fig. 4(b)) have already been introduced and represented in conceptual neighborhood graphs in [11] and [10], respectively.

Thus, to compute a spatio-temporal metric based on these conceptual neighborhood graphs, we propose to use a graph product between them. Fig. 5 presents a part of the graph product between Fig. 4(a) and Fig. 4(b).

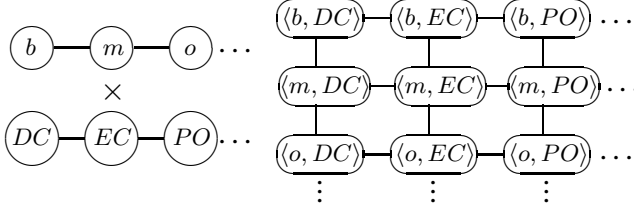


Fig. 5. Graph product between Fig. 4(a) and Fig. 4(b)

Thanks to the conceptual neighborhood graph of Fig. 5, a conceptual neighborhood distance δ between two spatio-temporal relations is computed from their shortest path distance. In this section, each branch is weighted by 1. We will see, in section 6, that this weight can be different on each arc depending on multimedia object properties.

To compute adapted relation graphs solutions (section 5.1), we propose to generate, with the Nebel backtracking algorithm [12], all consistent relation graphs from the relation graph corresponding to possible executions, and select those which are at the minimal distance of the relation graph corresponding to potential executions.

For example, Fig. 6 (left) is an adapted relation graph solution computed from the relation graph of Fig. 3 (right). We indicate on Fig. 6 (left) each conceptual neighborhood distance δ from the relations used in the initial document specification of Fig. 3 (left). Its global conceptual neighborhood distance d from Fig. 3 (left) is 6 ($3 + 3$) which is the minimal distance. Next to the adapted relation graph solution of Fig. 6, a corresponding possible execution is presented.

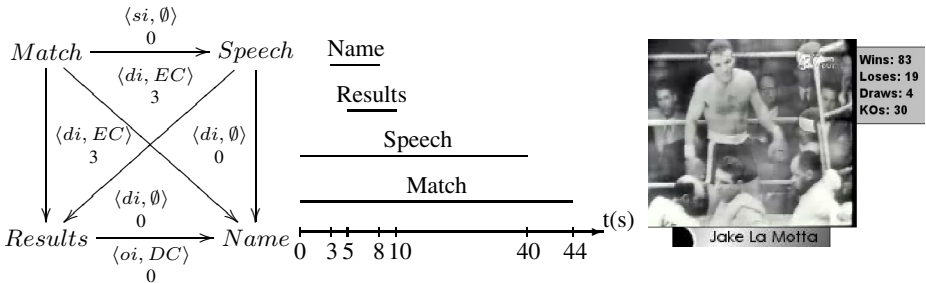


Fig. 6. An adapted relation graph solution (left) and one of its possible execution (right)

In the following section, animated documents (*i.e.*, moving documents) are considered. Thus, we extend both the spatio-temporal representation and the proximity metric.

5.3 Spatio-temporal Proximities for Animated Documents

A multimedia document can be composed of animations, *i.e.*, visible objects can move during their presentation. Hence, a spatio-temporal relation $r = \langle r_T, r_S \rangle$ between two multimedia objects can be composed of several spatial relations.

A straightforward encoding of the animation between two multimedia objects could consist of specifying r_S as an ordered list of spatial relations, *i.e.*, $r = \langle r_T, r_S^1 \rightarrow \dots \rightarrow r_S^n \rangle$. However, with this encoding, it is difficult to identify when a particular object hides another one at a time because we consider a global temporal relation and several spatial relations. For example, suppose Fig. 7 with $Results\{ \langle oi, DC \rightarrow EC \rightarrow PO \rangle \} Name$. It is difficult to identify if $Results PO Name$ (where $Results$ hides partially $Name$) at a time. If $Name$ is played from second 3 to 8 and $Results$ from second 5 to 10, do the objects hide partially between second 5 and 8 or not.

Thus, we propose an encoding that relies on a spatio-temporal partition of the animation. As an example, Fig. 7 presents this spatio-temporal partition.

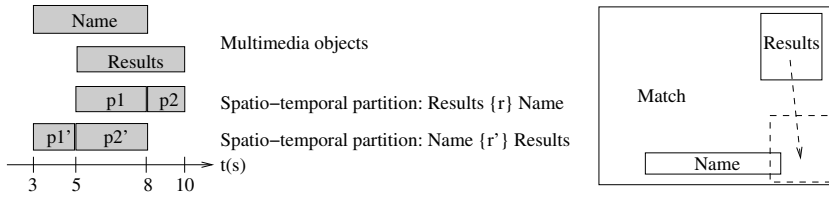


Fig. 7. The spatio-temporal partitions (left) describing the animation (right)

The global spatio-temporal relation r between two objects A and B is encoded by successive spatio-temporal relations denoting the animation, *i.e.*, $r = r_1 \rightarrow \dots \rightarrow r_n$ where n is the number of temporal interval of the spatio-temporal partition. Each temporal interval p_i of the partition is related to the first object A (e.g., *Results*) and has a spatio-temporal relation $r_i = \langle r_T, r_S \rangle$ with the second object B (e.g., *Name*). With this encoding, we have to ensure that each temporal interval of the partition p_{i+1} immediately follows in time p_i . Note that we also have to consider inverse relation in relation graphs because, with this encoding, it could happen that $(A \{r\} B) \neq (B \{(r)^{-1}\} A)$.

For example, suppose Fig. 7 with $Results\{ \langle f, EC \rangle \rightarrow \langle mi, PO \rangle \} Name$. There are two temporal intervals $p1$ and $p2$ in the spatio-temporal partition. The former specifies that $Results\langle f, EC \rangle Name$. This temporal interval is immediately following in time by the latter specifying that $Results\langle mi, PO \rangle Name$. In this case, it is possible to identify if $Results PO Name$ during they are played in parallel or not. Moreover, $r' \neq (r)^{-1}$, e.g., $Name\{ \langle m, DC \rangle \rightarrow \langle s, EC \rangle \} Results$ denoted by $p1'$ and $p2'$.

Computing spatio-temporal proximities over this kind of relations is quite similar to the approach explained in section 5.1. The conceptual distance d between relation graphs, as defined in Def. 2, is thus extended to take into account the list of spatio-temporal relations.

Definition 3 (Conceptual distance between spatio-temporal relation graphs).

$$d(\lambda, \lambda') = \sum_{n, n' \in N} \text{Min}_{r \in \lambda(\langle n, n' \rangle), r' \in \lambda'(\langle n, n' \rangle)} \sum_{\langle t, s \rangle \in r, \langle t', s' \rangle \in r'} \delta(\langle t, s \rangle, \langle t', s' \rangle)$$

Unfortunately, our spatio-temporal representation has limitations. Since we manipulate, in this paper, Allen relations, our spatio-temporal encoding of animations does not allow to identify spatial events during an instant, *e.g.*, rebounds. However, we can manage this problem by using other representations based on points and intervals, *e.g.*, [13], and defines proximities over these representations.

Spatio-temporal proximities between static and animated documents have been defined in the context of adapting multimedia documents. We propose in the next section to refine these proximities for providing appropriate adapted solutions according to multimedia object properties.

6 Refining the Spatio-temporal Proximities

A multimedia object have properties, such as its media type (*e.g.*, text, image, audio, video), its shape. . . The multimedia document adaptation should compute close adapted documents respecting adaptation constraints according to these properties.

As far as time is concerned, we could consider that texts and images are elastic objects, *i.e.*, we can easily stretch or reduce their presentation. On the other hand, audios and videos are rigid objects, *i.e.*, they have an intrinsic execution time. When adapting a multimedia document it could be beneficial to propose adapted solutions that only deform elastic object and preserve rigid ones.

Thanks to these multimedia object properties, spatio-temporal proximities can be refined by using appropriate conceptual neighborhood graphs. As explained in [11], conceptual neighborhood graphs have different structures (*e.g.*, Fig. 4(a) and Fig. 8(a)) depending on the types of deformation of the objects involved. Thus, defining different proximities between relations.

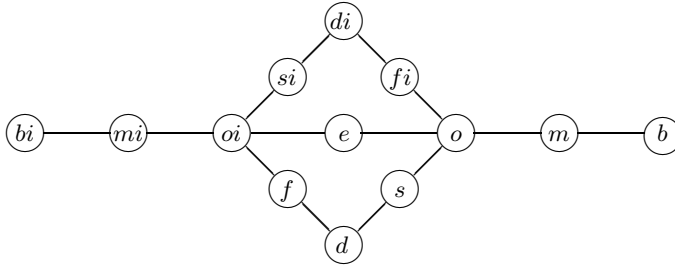
For example, if we consider two rigid objects A and B that are equal (e). The only way to transform the relation between them is to move in time the objects without changing their duration. Hence, having the relations overlaps (o) or overlapped-by (oi) neighbor of the relation equals (e). Fig. 8(a) presents the conceptual neighborhood graph of rigid objects.

On the other hand, if we consider two elastic objects A and B that are equal (e). Suppose, it is only possible to deform one extreme point of one of the two objects. The relations equals (e) and overlaps (o) are not neighbor anymore and other relations are neighbor, *e.g.*, starts (s) or finishes (f) (Fig. 4(a)).

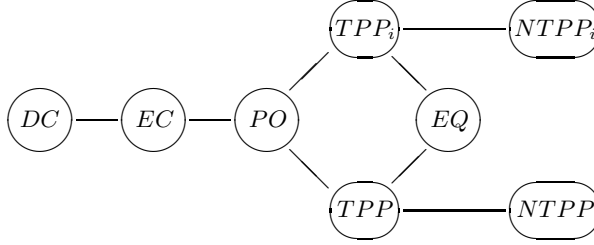
In a similar way, spatial conceptual neighborhood graphs can have different structures (Fig. 4(b) and Fig. 8(b)). For example, Fig. 4(b) considers circles deformation, while Fig. 8(b) considers rectangles deformation (if we suppose that only one rectangle can be deform horizontally or vertically).

Note that when two objects are compared with different properties, *e.g.*, a rigid and an elastic object, we propose to merge the two conceptual neighborhood graphs in a single one, *i.e.*, adding all neighborhood relationships used in both neighborhood graphs.

Moreover, we have consider in section 5 that all neighbor relations have a distance $\delta = 1$. Conceptual neighborhood graph branches can be weighted differently. For example, suppose two images that overlaps in time. Suppose, they have such property that



(a) Conceptual neighborhood graph of Allen relations.



(b) Conceptual neighborhood graph of RCC8 relations.

Fig. 8. Other conceptual neighborhood graphs

they are related to each other. It is more appropriate to display these images together, hence having the weight between the relation overlaps (o) and the neighbor relations starts (s) and finished-by (fi) greater than the neighbor relation meets (m) (even if the relations s , fi and m are neighbors of o , Fig. 4(a)).

All these conceptual neighborhood graph structures imply different structures of the spatio-temporal neighborhood graph (Fig. 5). Hence, refining the spatio-temporal proximities and the multimedia document adaptation.

7 Conclusion

Combining the temporal and spatial dimensions of multimedia documents, we define proximities between spatio-temporal relations. These proximities are used in a semantic approach for adapting multimedia documents along their spatio-temporal dimension, thus providing adapted documents close to the initial one respecting adaptation constraints. Moreover, we extend the spatio-temporal representation and the metrics to take into account animations. Finally, we refine the spatio-temporal proximities according to multimedia object properties.

In the future, we plan to apply the adaptation framework to other dimensions, especially the logical and interactive ones. We thus have to define new metrics denoted proximities between multimedia documents. Moreover, we also want to apply the spatio-temporal adaptation approach to standard multimedia description languages, *e.g.*, SMIL. As proposed in [2], translation functions from SMIL documents to the spatio-temporal representation and vice versa have to be specified to apply the semantic adaptation approach.

References

1. Euzenat, J., Layaïda, N., Dias, V.: A semantic framework for multimedia document adaptation. In: Proc. of IJCAI'03, Morgan Kauffman (2003) 31–36
2. Laborie, S., Euzenat, J., Layaïda, N.: Adaptation spatiale efficace de documents SMIL. In: 15e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle (RFIA), Tours (France) (2006)
3. Roisin, C.: Authoring structured multimedia documents. In: Conference on Current Trends in Theory and Practice of Informatics. (1998) 222–239
4. W3C: Synchronized Multimedia Integration Language (SMIL 2.0) Specification. (2001) <http://www.w3.org/TR/smil20/>.
5. Jourdan, M., Layaïda, N., Roisin, C., Sabry-Ismaïl, L., Tardif, L.: Madeus, an authoring environment for interactive multimedia documents. In: 6th ACM Multimedia conference, Bristol (UK) (1998) 267–272
6. Marriott, K., Moulder, P., Stuckey, P.J., Borning, A.: Solving disjunctive constraints for interactive graphical applications. *Lecture Notes in Computer Science* **2239** (2001) 361–376
7. W3C: Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies. (2001) <http://www.w3.org/TR/CCPP-struct-vocab/>.
8. Lemlouma, T., Layaïda, N.: The negotiation of multimedia content services in heterogeneous environments. In: Proc. 8th International Conference on Multimedia Modeling (MMM01), Amsterdam (NL) (2001) 187–206
9. Allen, J.: Maintaining knowledge about temporal intervals. *Communications of the ACM* **26**(11) (1983) 832–843
10. Randell, D.A., Cui, Z., Cohn, A.: A spatial logic based on regions and connection. In: Nebel, B., Rich, C., Swartout, W., eds.: KR'92. Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference, San Mateo, California, Morgan Kaufmann (1992) 165–176
11. Freksa, C.: Temporal reasoning based on semi-intervals. *Artificial Intelligence* **54**(1–2) (1992) 199–227
12. Nebel, B.: Solving hard qualitative temporal reasoning problems: Evaluating the efficiency of using the ORD-horn class. In: European Conference on Artificial Intelligence. (1996) 38–42
13. Ma, J., Knight, B., Petridis, M.: A revised theory of action and time based on intervals and points. *The Computer Journal* **37**(10) (1994) 847–857

Deep into Color Names: Matching Color Descriptions by Their Fuzzy Semantics

Haiping Zhu, Huajie Zhang, and Yong Yu

Department of Computer Science and Engineering, Shanghai Jiao Tong University,
Shanghai, 200240, P.R. China
{zhu, zhjay, yyu}@apex.sjtu.edu.cn

Abstract. In daily life, description and location of certain objects on the web are much dependent on color names. Therefore, a maturely-implemented matching subsystem for color descriptions will certainly facilitate web applications in the domains concerned, such as image retrieval, clothing search, etc. However, both keyword matching and semantic mediation by the current ontologies are confronted with difficulties in precisely evaluating the similarity between color descriptions, which requests the exploitation of “deeper” semantics to bridge the semantic gap. What with the inherent variability and imprecision characterizing color naming, this paper proposes a novel approach for defining (1) the fuzzy semantics of color names on the HSL color space, and (2) the associated measures to evaluate the similarity between two fuzzified color descriptions. The experimental results have preliminarily shown the strength of the deeper semantics surpassing the ability of both keywords and WordNet, in dealing with the matching problem of color descriptions.

Keywords: Matching Color Descriptions, HSL Color Space, Fuzzy Colors, Fuzzy Color Similarity, Color Difference Formula.

1 Introduction

Color is one of the features that the human brain memorizes the most [1]. People rely heavily on color names to describe and locate certain objects, e.g. color images [2, 3], flowering plants [4, 5], clothing commodities, etc. Therefore, a maturely-implemented matching subsystem for color descriptions will certainly facilitate web search applications in the domains that are concerned. However, by traditional keyword matching, it is hard to predict that “*crimson*” and “*ruby*” share almost the same meaning as “*deep red*”. Subsequently, the emergence of semantic mediation techniques, such as the adoption of an ontology (e.g. WordNet [6]), helps people to unify the semantics of color terms by concepts (e.g. *synsets* in WordNet). Nevertheless, it is still difficult for such an ontology to tell the difference among color descriptions like “*Turkey red*”, “*deep red*” and “*orange red*”, because all of them are immediate subconcepts of “*red*” and there is no more information provided to distinguish them. The ultimate way we believe to tackle this issue is to match color descriptions by “deeper” semantics, the semantics defined on certain color space models.

On color spaces, *color difference formulae* are most often used to define the similarity between two colors (a rather comprehensive review can be found in [7]).

However, a multiplicity of uncertainties are presented in color descriptions mainly because of the subjective and context-sensitive nature of color [8]. Hence, it is quite unlikely that different people would map the same color name to a single color point in different applications. To achieve shared understanding for semantic integration, it asks for the incorporation of fuzzy semantics of colors in similarity evaluation.

Gärdenfors claims that color terms correspond to convex regions on the color space [9]. Similar thoughts can be found in separate researches by [3] and [5] in which the semantics of each color term is represented by a range triplet on the three dimensions of a chosen color space. In our opinion, however, the range triplets fail to reflect the intuition that the central points of the regions would be more representative than the peripheral ones. Fuzzy colors, i.e. colors that are defined on fuzzy sets, have been well researched in the academia of CBIR (Content Based Image Retrieval) [10, 2]. With the aid of membership functions, fuzzy color similarity can be evaluated. However, in CBIR, it is the similarity between two color points (e.g. colors of pixels from two images) that is primarily investigated, through their membership grades to several designated fuzzy colors, respectively. Differently, when we match color descriptions (other than pixels in images), the number of fuzzy colors in the system is arbitrary. Furthermore, two color regions, denoted by two color descriptions, respectively, are actually being compared, which requests the design of new similarity measures.

Targeting the above objectives, this paper proposes a new approach for matching color descriptions, featuring:

1. **The membership function for an arbitrary fuzzy color**, either *achromatic* (i.e. “black”, “white” and all shades of “gray”) or *chromatic* (the opposite to *achromatic*), as the composite of all the three component membership functions on the HSL color space (a prominent perceptual color space, see Section 2 for more details);

2. **The evaluation of similarity between fuzzified color descriptions**, according to the membership functions defined, for different combinations of the queried color and the color in the resource repository (hereinafter “the resource color”) belonging to the two categories of the chromatic and the achromatic.

The experimental results have preliminarily shown the strength of the deeper semantics surpassing the ability of both keywords and WordNet, in dealing with the matching problem of color descriptions.

The remainder of this paper is organized as follows. Section 2 briefly introduces the color spaces we are concerned. The fuzzified representation of color semantics is defined in Section 3, while the similarity between two fuzzy colors defined in Section 4. Section 5 presents a brief introduction of the prototype system and an analysis of the preliminary experimental results. Related work is discussed in Section 6. Section 7 concludes the whole paper and gives several interesting topics for further research.

2 Color Spaces

The most commonly used color space to represent colors on web pages or printing materials may be RGB. In this space, each color is represented by three independent dimensions, namely *Red* (abbr. *R*), *Green* (abbr. *G*) and *Blue* (abbr. *B*). Usually, the value on each dimension falls into the range between 0 and 255. For example, “black”

is assigned (0, 0, 0), “white” is assigned (255, 255, 255), and all shades of “gray” is located on the diagonal from “black” to “white”, characterized as $R = G = B$. The other points in the cube depict chromatic colors (Figure 1(a)).

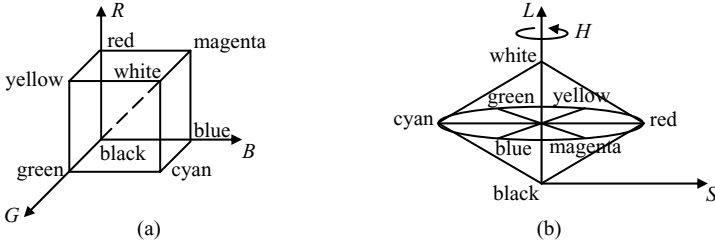


Fig. 1. The two color space models concerned in this paper: (a) RGB; (b) HSL

Despite its popular application in human life, RGB is also widely accepted as a color space not perceptually uniform to human vision [2, 5, 7, 10]. This paper focuses on color representation defined on the HSL space [11] which is believed to be close to the physiological perception of human eyes, and furthermore possesses easy-and-clear transformation from the RGB space (refer to [10]) as well as a conveniently realizable color difference formula (see equation (7) in Section 4.1). However, we also believe that our design rationale can be transferred to other perceptual or perceptually uniform color spaces by meticulous definition.

The HSL space tries to decompose colors according to physiological criteria as *Hue*, *Saturation* and *Luminance*. Hue (abbr. H) refers to the pure spectrum colors and corresponds to dominant colors as perceived by a human. It is an angle that takes a value between 0 and 360. Saturation (abbr. S) corresponds to the relative purity or the quantity of white light that is mixed with hue, while luminance (abbr. L) refers to the amount of light in a color. Both of them are in the form of ratio and are thus within the interval of $[0, 1]$. Figure 1(b) depicts the HSL color model. The points on the L -axis with H undefined and $S = 0$ denote achromatic colors, while the remaining the chromatic.

3 Representation of Fuzzy Colors

In the literatures concerning fuzzy colors [10, 2], trapezoidal membership function is usually employed to model each separate dimension of a fuzzy color (Figure 2):

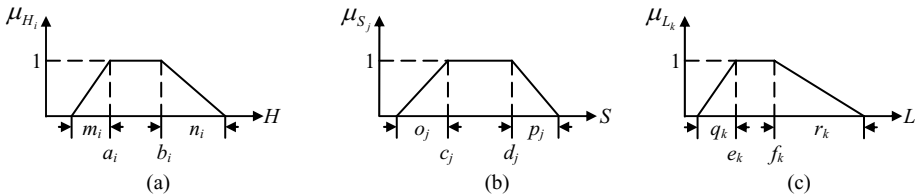


Fig. 2. The membership degree of the fuzzy (a) H ; (b) S ; (c) L

$$\mu_{H_i}(h) = \begin{cases} 1, & a_i \leq h \leq b_i \\ 1 - \frac{a_i - h}{m_i}, & a_i - m_i < h < a_i \\ 1 - \frac{h - b_i}{n_i}, & b_i < h < b_i + n_i \\ 0, & \text{otherwise} \end{cases} \quad \mu_{S_j}(s) = \begin{cases} 1, & c_j \leq s \leq d_j \\ 1 - \frac{c_j - s}{o_j}, & c_j - o_j < s < c_j \\ 1 - \frac{s - d_j}{p_j}, & d_j < s < d_j + p_j \\ 0, & \text{otherwise} \end{cases} \quad \mu_{L_k}(l) = \begin{cases} 1, & e_k \leq l \leq f_k \\ 1 - \frac{e_k - l}{q_k}, & e_k - q_k < l < e_k \\ 1 - \frac{l - f_k}{r_k}, & f_k < l < f_k + r_k \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

After the fuzzy sets of S and L are defined, we use them to compose the fuzzy set of *tone* (denoted as T), i.e. the S - L plane. After investigating the set of transformation formulae from RGB to HSL [10], we can observe that the definition of S is dependent on L , which forms a triangular area (Figure 3(a)). In other words, the composite result of S and L is situated inside this triangular area, while the outside never appears in this color system. Thus, according to the operation of the *algebraic product* (enlightened by [10]) and associated with a two-valued function f_T , the fuzzy set of T can be defined as:

$$\mu_{T_k}(s, l) = \mu_{S_j}(s) \cdot \mu_{L_k}(l) \cdot f_T(s, l) \quad (2)$$

$$f_T(s, l) = \begin{cases} 1, & \text{if } s \leq \frac{l}{0.5} \text{ when } l \leq 0.5 \text{ or } s \leq \frac{1-l}{0.5} \text{ when } l > 0.5 \\ 0, & \text{otherwise} \end{cases}$$

We further define the membership function for the fuzzy set of a chromatic color as the algebraic product of the membership grade of H and that of T .

$$\mu_{CC_x}(h, s, l) = \mu_{H_i}(h) \cdot \mu_{T_k}(s, l) \quad (3)$$

Now, each chromatic color CC_x is mapped to a cake-like closed region on the HSL space with μ_{CC_x} as the membership function defining the fuzzy set on the “cake” (Figure 3(b)).

As to an achromatic color, its H is undefined and $S = 0$, so its membership function is measured only by the membership grade of L .

$$\mu_{AC_y}(l) = \mu_{L_k}(l) \quad (4)$$

Each achromatic color AC_y is mapped to a line segment located on the L -axis of the HSL space with μ_{AC_y} as its membership function (Figure 3(b)).

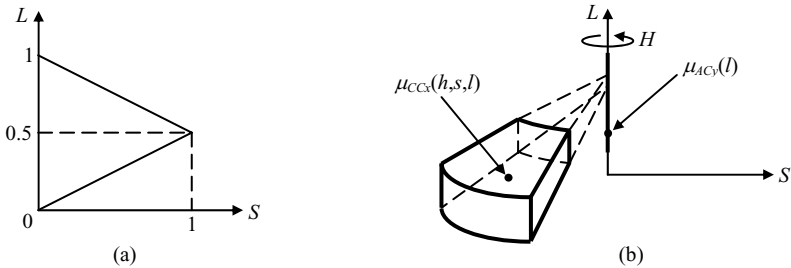


Fig. 3. The representation of fuzzy colors: (a) the tone plane; (b) a chromatic color – a “cake”, and an achromatic color – a line segment

4 Similarity Between Color Descriptions

In the field of CBIR, the crisp color of each pixel in an image can be represented by its membership degrees to a set of fuzzy colors, and then the similarity between two colors is determined by these membership degrees. A typical fuzzy similarity measure is defined as:

$$Sim(\tilde{C}_1, \tilde{C}_2) = \frac{\sum_{i=1}^h \sum_{j=1}^r \min(\mu_{C_{ij}}(h_1, l_1, s_1), \mu_{C_{ij}}(h_2, l_2, s_2))}{\sum_{i=1}^h \sum_{j=1}^r \max(\mu_{C_{ij}}(h_1, l_1, s_1), \mu_{C_{ij}}(h_2, l_2, s_2))} \quad (5)$$

where \tilde{C}_1 and \tilde{C}_2 are the fuzzy representation of two crisp colors $C_1 = (h_1, l_1, s_1)$ and $C_2 = (h_2, l_2, s_2)$, respectively, with respect to a group of fuzzy colors C_{ij} with $\mu_{C_{ij}}$ as their membership functions [10].

As mentioned in Section 1, in our proposed approach, the comparison is not between two color points (e.g. \tilde{C}_1 and \tilde{C}_2 in equation (5)), but between two color descriptions (e.g. two different C_{ij} 's), namely two sets of color points. Taking the hue dimension as an example (though the three dimensions of the color space are *integral* [9] and should actually be considered as a whole during color matching), we exemplify as follows how the similarity between color descriptions is measured (Figure 4).

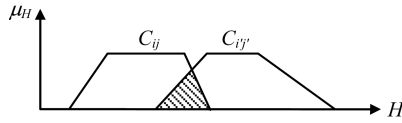


Fig. 4. The comparison between two color descriptions (on the hue dimension)

Intuitively, the conjunction (overlap) of two colors is taken as a certain region on the color space that can be said to be either of the two colors. If they are identical, the overlap reaches its maximum (i.e. the same in size as each of them); when they are disjoint (e.g. two complementary colors), the overlap is the minimum (i.e. zero), which means they share no object at all. Hence, we believe that the overlap of two colors on the color space can be adopted to evaluate how similar they are to each other.

It should also be noted that in fact we define the similarity measure as asymmetric: we are measuring how much the resource color matches (namely, is subsumed by) the queried one, but not the reverse. Therefore, the similarity measure is given the form as the ratio of the overlap to the resource color.

In the following text, we explore the discussion of our fuzzy color similarity by three categories: chromatic color vs. chromatic color, achromatic color vs. achromatic

color and chromatic color vs. achromatic color, according to the different characteristics between the two types of colors shown in Section 3.

4.1 Chromatic Color vs. Chromatic Color

As analyzed above, if two chromatic colors overlap on the color space, the degree to which they overlap each other can be used to measure their similarity. Otherwise, we turn to investigate the distance between them: taking the closest two points selected from the two fuzzy colors respectively, we regard the distance between them as the minimum cost of *rotation* and/or *move* to make the two fuzzy colors overlap.

1. Derived from the definition in Section 3, only when two chromatic colors have overlap on each dimension will they overlap on the color space (Figure 5). Special attention should be paid to the periodicity of H because it is an angle.

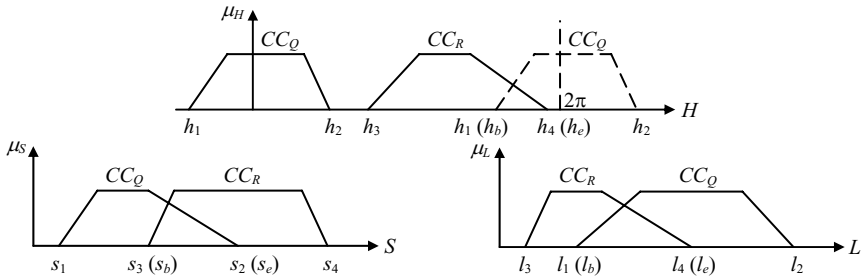


Fig. 5. Two chromatic colors have overlap on each dimension

In this case, we define the *overlap* between the two colors as the ratio of two integrals in the cylindrical coordinate system with s as the radius, h the angle and l the height:

$$overlap(CC_Q, CC_R) = \frac{\int_{l_b}^{l_e} dl \int_{h_b}^{h_e} dh \int_{s_b}^{s_e} \min\{\mu_{CC_Q}(h, s, l), \mu_{CC_R}(h, s, l)\} \cdot s ds}{\int_{l_3}^{l_4} dl \int_{h_3}^{h_4} dh \int_{s_3}^{s_4} \mu_{CC_R}(h, s, l) \cdot s ds} \quad (6)$$

where CC_Q and CC_R are the queried chromatic color and the resource chromatic color, with μ_{CC_Q} and μ_{CC_R} as their membership functions, respectively. The integral intervals $[h_b, h_e]$, $[s_b, s_e]$, $[l_b, l_e]$ in the numerator are the intervals where the two colors overlap on the H , S , L dimensions, respectively. The integral intervals $[h_3, h_4]$, $[s_3, s_4]$, $[l_3, l_4]$ in the denominator are determined by the boundary of the resource color. If we take μ_{CC} as the function of “density”, equation (6) describes the ratio of the minimum “mass” of the closed region where the two colors overlap to the “mass” of the resource color. The value of *overlap* ranges between 0 (either of $[h_b, h_e]$, $[s_b, s_e]$ and $[l_b, l_e]$ is reduced to a point) and 1 (CC_R is just the same as CC_Q).

2. If two chromatic colors have no overlap on each dimension (Figure 6):

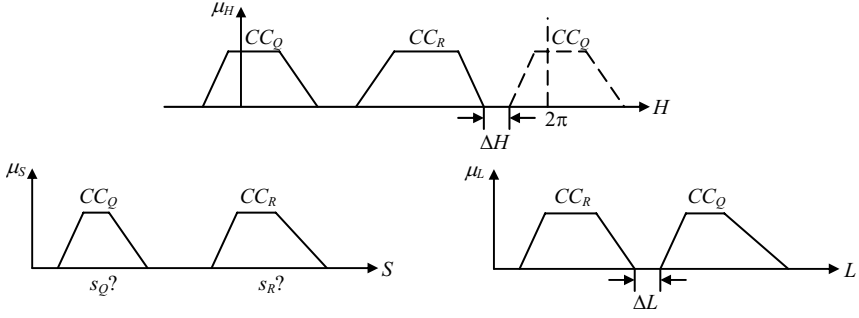


Fig. 6. Two chromatic colors have no overlap on each dimension

We turn to use the color difference formula on the HSL space to define the *distance* between the two colors:

$$distance(CC_Q, CC_R) = \sqrt{\Delta L^2 + s_Q^2 + s_R^2 - 2s_Q s_R \cos(\Delta H)} \quad (7)$$

where ΔH and ΔL are defined as the distance on H and L , respectively, between the closest two points. Since it is not ΔS but the values of s_Q and s_R that affect the calculation of *distance*, we let $(s_Q, s_R) = \arg \min_{\text{all } s \text{ in } CC_Q, CC_R} \{s_Q^2 + s_R^2 - 2s_Q s_R \cos(\Delta H)\}$.

It is not difficult to conclude that the minimum of *distance* is 0, while the maximum is 2, achieved when two saturated and complementary colors ($\Delta L = 0$, $s_Q = s_R = 1$, $\Delta H = 180$) are being compared. What's more, the *distance* measure is symmetric.

3. If two chromatic colors have overlap on either, but not all, of the dimensions (i.e. the combination of separate cases on H , S , L in 1 and 2), we calculate the *distance* according to equation (7) by designating $\Delta L = 0$ for overlap on L , $\Delta H = 0$ for overlap on H , and the valuation of (s_Q, s_R) in the same way as discussed above.

Because the more distance two colors hold the less similarity they will have, we use the value of the opposite number of *distance* to measure their similarity. Thus, the similarity between two chromatic colors takes a value between -2 and 1 (from *distance* to *overlap*).

4.2 Achromatic Color vs. Achromatic Color

Achromatic colors have normal membership functions on the L dimension only. Therefore, determining the similarity between two achromatic colors is reduced to the measurement of their *overlap* or *distance* on L :

$$overlap(AC_Q, AC_R) = \frac{\int_{l_b}^{l_e} \min\{\mu_{AC_Q}(l), \mu_{AC_R}(l)\} dl}{\int_{l_3}^{l_4} \mu_{AC_R}(l) dl} \quad (8)$$

$$distance(AC_Q, AC_R) = |\Delta L| \quad (9)$$

where AC_Q and AC_R are the queried achromatic color and the resource achromatic color, with μ_{AC_Q} and μ_{AC_R} as their membership functions, respectively. The definition of other parameters is the same as in Section 4.1. The value *overlap* measures the ratio of the area shared by the two achromatic colors to that of the resource color, while the value *distance* measures the distance between the closest two points selected from the two achromatic colors, respectively.

The maximum of *overlap* is 1 when the two achromatic colors are totally the same, while the maximum of *distance* is also 1 (at the moment, the similarity is -1) when the two achromatic colors are just “black” and “white”. When the two achromatic colors share only one color point, both *overlap* and *distance* reach their minimum 0. Therefore, the similarity between two achromatic colors ranges between -1 and 1.

4.3 Chromatic Color vs. Achromatic Color

Since each achromatic color is characterized as $S = 0$, the integral interval $[s_b, s_e]$ defined in Section 4.1 is either the point $S = 0$ or \emptyset (i.e. no overlapping). Therefore, calculating the *overlap* between a chromatic color and an achromatic color always returns the result 0. Hence, we only observe the distance between such two colors. Because the *distance* measure is symmetric, we do not distinguish whether the achromatic color is the queried one or the resource one.

Because the H of each achromatic color is undefined, we may take $\Delta H \equiv 0$ when it is compared with a chromatic color (let the H of the achromatic color equal to that of the chromatic color). Thus, the distance is based on the tone plane only (Figure 7):

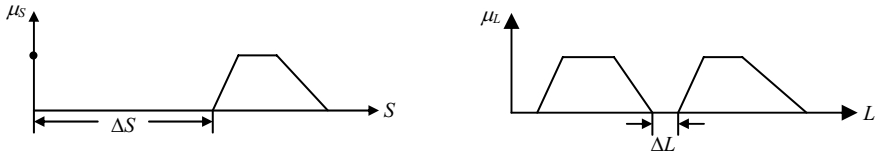


Fig. 7. The distance between a chromatic color and an achromatic color

$$distance(CC, AC) = \sqrt{\Delta L^2 + \Delta S^2} \quad (10)$$

It takes a value between 0 and $\sqrt{5}/2$ (when the chromatic color is a saturated one and the achromatic color is “black” or “white”), so the similarity, as its opposite, ranges between $-\sqrt{5}/2$ and 0.

5 Preliminary Experiments

We collected color names together with the RGB values of the sampling points for each color from “The Mother of All HTML Colors”¹, which can be taken as an open and collaborative color database. There were more than 11,650 entries of (*Color*

¹ <http://tx4.us/moacolor.htm>

Name(s), *RGB*) pairs gathered. After all the RGB values were transformed into HSL values, multiple colors belonging to the same entry were split. For the fact that people seldom write query as “gray 1” or “57% vivid cold blue” (57% refers to its luminance), ordinal numbers and percentages in color names were removed. Finally, entries with identical names but different HSL values were merged to construct the kernel of the membership function (i.e. the region in which the membership degree is 1) for that color. As to the determination of the interval between the kernel and the border where the membership degree falls to 0, we use the heuristic that it equals to a fixed proportion to the length of the kernel (e.g. 10% in our current implementation). If the kernel is reduced to a single point, we specify the interval as the minimum of those for non-single-point kernels. We believe in this way it to some extent models the generalization-specialization relationships between colors, because more general color terms tend to possess longer kernels. All the above steps eventually led to about 9,940 different fuzzy colors after their membership functions were defined.

Here we present an example query “dark red” together with its results in a keyword-based retrieval system, an ontology-based retrieval system and our prototype system, respectively. In the keyword-based system we implemented over the same color database, wildcard “*” is supported, and we assume the similarity returned is determined by the position where the wildcard occurs in a color description, i.e.: “*dark red” > “dark * red” > “dark red *” (intuitively, “*dark red” is still a certain kind of “dark red”, while “dark * red” is usually a kind of “red” with dark tone and “dark red *” could even be a non-red color dependent on what “*” represents). Hence, the ranking of the retrieved color names is as follows:

“dark red” > “somewhat dark red”, “very dark red” > “dark brownish red”, “dark carmine red”, “dark dull red”, “dark faded red”, “dark grayish red”, “dark hard red”, “dark Indian red”, “dark lacquer red”, “dark mineral red”, “dark purplish red”, “dark weak red”, “dark yellowish red” > “dark red brown”, “dark red orange”, “dark red violet”

In the ontology-based system, we adopt the similarity measure introduced in [12] to rank the concepts related to “dark red” in WordNet 2.1² in the order as: subconcepts > the superconcept > siblings’ subconcepts (the principle behind this ranking order is that each subconcept of “dark red” is a “dark red” as well, but the *semantic distance* along the concept hierarchy continually increases when it travels from “dark red” to its superconcept and further to its siblings, and finally to its siblings’ subconcepts). With the aid of hyponymy in WordNet, we obtain the ranking result as follows (each pair of braces denotes a synset in WordNet):























































{dark red}, {burgundy}, {claret}, {oxblood red}, {wine, wine-colored} > {red} > {sanguine}, {chrome red}, {Turkey red, alizarine red}, {cardinal, carmine}, {crimson, ruby, deep red}, {purplish red}, {cerise, cherry, cherry red}, {scarlet, vermilion, orange red} > {fuchsia, magenta}, {maroon}

In our prototype system, the similarity between colors is calculated by *overlap* or *distance* on the HSL space. Table 1 depicts a part of the top-ranked (rank#1–30) color names and the ranking of the color names that are concerned in the above two systems

² <http://wordnet.princeton.edu/>

(except “*chrome red*” which cannot be found in the color database). It should be noted that here we skip some top-ranked color names, such as “*overcooked sausage*”, “*Emergency!...Emergency!!*”, etc., to make the result set more normalized. Each color cell corresponds to a defuzzification (e.g. *Center of Gravity*, namely the geometrical center of gravity of a given area, in our implementation, which is characterized by Gärdenfors as the most prototypical example [9]) of a certain fuzzy color, with the leftmost top as the reference color, i.e. “*dark red*”.

Table 1. The retrieval result of the example query “*dark red*” in our prototype system

																	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
																	
19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
																	
37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52		

ID	Name	Rank	ID	Name	Rank	ID	Name	Rank
1	dark red	#1	19	Turkey red	#53	37	dark red orange	#362
2	carmine red	#2	20	red	#65	38	maroon	#448
3	dark brownish red	#7	21	dark grayish red	#73	39	oxblood red	#471
4	gaillardia red	#8	22	cherry	#74	40	dark hard red	#492
5	brownish carmine	#10	23	cherry red	#76	41	orange red	#525
6	brownish red	#11	24	dark dull red	#77	42	claret	#579
7	Spanish red	#12	25	carmine	#108	43	dark purplish red	#588
8	vivid brown red	#13	26	crimson	#113	44	wine	#690
9	dark cherry	#14	27	deep red	#125	45	dark weak red	#751
10	tomato red	#16	28	ruby	#138	46	burgundy	#878
11	dark carmine red	#17	29	sanguine	#149	47	dark Indian red	#1115
12	Chimayo red	#19	30	vermillion	#154	48	dark mineral red	#1115
13	vivid lilac red	#20	31	scarlet	#177	49	dark red brown	#1160
14	dark faded red	#24	32	somewhat dark red	#189	50	magenta	#1282
15	Sultan red	#25	33	dark yellowish red	#239	51	fuchsia	#1389
16	alizarine red	#29	34	cerise	#247	52	dark red violet	#2002
17	cardinal	#30	35	purplish red	#275			
18	very dark red	#34	36	dark lacquer red	#322			

From Table 1, it can be observed that though keyword matching produces some good results, e.g. “*dark brownish red*”, “*dark carmine red*”, etc., it fails to retrieve many other color names that are perceptually, but not literally, similar to “*dark red*” (refer to the top-ranked color names in our system and their corresponding color cells) and it cannot avoid to retrieve irrelevant color names such as “*dark red violet*”. In the result set given by the ontology (WordNet), though its coverage on the color database is rather low, the siblings of “*dark red*” contain quite a few concepts that are literally distinct from but semantically close to it, e.g. “*alizarine red*”, “*cardinal*”, etc. Unfortunately, the same collection also includes very irrelevant terms such as “*orange red*” (the reader may compare its color cell with the reference color). It might be argued that WordNet is a generic lexicon and thus not particularly designed for the color domain. However, we believe it is already enough for one to learn from the experiment that merely adopting the concept hierarchy is not sufficient to determine color

similarity. Besides, we are planning to introduce the Getty Art and Architecture Thesaurus [13], an authoritative online color resource, and the CSIRO color ontology [14] if available online, which contains 12,471 classes together with “proper part of” relations between classes, to improve the quality of the validation experiment.

From the above observations, we can conclude that color names ask for more perceptual semantic representation than keywords and the current ontologies, and our proposed approach based on fuzzy colors on the HSL space preliminarily shows its strength in bridging this semantic gap. Currently, we are integrating the prototype system into a clothing search system to further evaluate its performance in a real application.

6 Related Work

Gärdenfors philosophically conjectures in [9] that colors are *natural properties* structured as convex regions in a conceptual space. Similar to this notion, Wang *et al* [5] represent the semantics of each color term by a cuboid space, defined by a range triplet on the three component measures of the HSL color model. However, they only use a color reasoner to perform subsumption checking, rather than similarity ranking. Liu and her colleagues [3] define a color naming model that also uses the combinations of value ranges on the three dimensions of the HSV color space (one that is very similar to HSL). Nevertheless, we think such range triplets are too rough-grained to precisely capture the fuzzy semantics of a color description.

[2] indexes dominant color(s) of an image by a tree of classes consisting of *fundamental colors/non-colors* (H) and their *colorimetric qualifiers* (S and L) on the HSL space. Though fuzzy representation is concerned, they perform subsumption checking other than similarity measuring, so, like [5], the ranking of retrieved instances is not supported. In [10], membership functions of a set of fuzzy colors based on the HSL color space are first defined to represent the membership degrees of each pixel in an image to these fuzzy colors. Then, a fuzzy similarity measure is developed for evaluating the similarity of fuzzy colors between two pixels. However, their work does not address achromatic colors. Moreover, our approach focuses on matching color names, which can be taken as collections of color points, other than matching pixels.

7 Conclusion

The conciliation of color semantics is crucial to the information integration from different data sources concerning color-related domains. Though color similarity measures are well researched in the realm of CBIR, little work is done on how to match two color descriptions while overcoming the defects in the current keyword-matching and ontology-mediation techniques. In this paper, we propose a novel approach to define a unified membership function on the three dimensions of the HSL color space for both chromatic and achromatic colors, and a set of similarity measures to evaluate how well two color descriptions match each other. The experimental results have preliminarily shown the strength of our approach in matching color descriptions by exploiting their fuzzy semantics.

There is still a lot of future work to follow. Composite colors are not rare in the derivation of colors, so we need to elicit membership functions for the descriptive

patterns as “*Color_X-ish Color_Y*”, “*Color_X* with touches of *Color_Y*” and so on. The proposed approach can be further assessed using quantified criteria, such as the precision-recall evaluation, in real applications. Last but not least, the topic of this paper has shown that numerical processing can be helpful to certain semantic harmonization tasks that are beyond the capability of keywords and the current ontologies. We are expecting the experiments on other clothing features, such as size/length, material ingredient, etc., to further verify this thought.

References

1. Shoemaker, S.: Colors, Subjective Relations, and Qualia. *Philosophical Issues*, 7 (1996) 55–66
2. Younes, A.A., Truck, I., Akdag, H.: Color Image Profiling Using Fuzzy Sets. *Turk. J. Elec. Engin.*, 13(3) (2005) 343–359
3. Liu, Y., Zhang, D., Lu, G., Ma, W.-Y.: Region-Based Image Retrieval with High-Level Semantic Color Names. In Chen, Y.-P.P. (ed.): *Proc. of the 11th Intl. Conf. on Multi Media Modeling (MMM 2005)* (2005) 180–187
4. Das, M., Manmatha, R., Riseman, E.M.: Indexing Flowers by Color Names using Domain Knowledge-driven Segmentation. In: *Proc. of the 4th IEEE Workshop on Applications of Computer Vision (WACV'98)* (1998) 94–99
5. Wang, S., Pan, J.Z.: Ontology-Based Representation and Query of Colour Descriptions from Botanical Documents. In: Meersman, R., Tari, Z. (eds.): *Proc. of CoopIS/DOA/ODBASE 2005*, LNCS 3761. Springer-Verlag (2005) 1279–1295
6. Fellbaum, C. (ed.): *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge MA (1998)
7. Sarifuddin, M., Missaoui, R.: A New Perceptually Uniform Color Space with Associated Color Similarity Measure for Content-Based Image and Video Retrieval. In: *Proc. of ACM SIGIR 2005 Workshop on Multimedia Information Retrieval (MMIR 2005)* (2005) 1–8
8. Binaghi, E., Ventura, A.D., Rampini, A., Schettini, R.: A Knowledge-Based Environment for Assessment of Color Similarity. In: *Proc. of the 2nd IEEE Intl. Conf. on Tools for Artificial Intelligence* (1990) 768–775
9. Gärdenfors, P.: *Conceptual Spaces: The Geometry of Thought*. MIT Press (2000)
10. Chien, B.-C., Cheng, M.-C.: A Color Image Segmentation Approach Based on Fuzzy Similarity Measure. In: *Proc. of the 2002 IEEE Intl. Conf. on Fuzzy Systems (FUZZ-IEEE'02)* (2002) 449–454
11. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*. 2nd edn. Prentice Hall (2002)
12. Zhong, J., Zhu, H., Li, J., Yu, Y.: Conceptual Graph Matching for Semantic Search. In: Priss, U., Corbett, D., Angelova, G. (eds.): *Proc. of the 10th Intl. Conf. on Conceptual Structures (ICCS 2002)*, LNCS 2393. Springer-Verlag (2002) 92–106
13. The Getty Research Institute: Art and Architecture Thesaurus Online.
http://www.getty.edu/research/conducting_research/vocabularies/aat/
14. Lefort, L., Taylor, K.: Large Scale Colour Ontology Generation with XO. In: Meyer, T., Orgun, M.A. (eds.): *Proc. of the Australasian Ontology Workshop (AOW 2005)*, *Conferences in Research and Practice in Information Technology (CRPIT)*, Vol. 58. Australian Computer Society (2005) 47–52

Developing an Argumentation Ontology for Mailing Lists

Colin Fraser, Harry Halpin, and Kavita E. Thomas*

School of Informatics
University of Edinburgh

colinf@inf.ed.ac.uk, hhalpin@ibiblio.org, kavita.e.thomas@gmail.com

Abstract. Managing emails from list-servs is an open problem that we believe may be partially resolved by the introduction of a principled, argumentation-based approach towards their representation. We propose an argumentation ontology, called “Argonaut,” which has been developed for the domain of standards-driven W3C mailing lists. We use the extensible nature of RDF to fuse an argumentation-based approach with one grounded in an issue-management system. We motivate our ontology with reference to the domain and propose future work in this area.

1 Introduction

Anyone who is a member of a list-serv will know the problems that arise when trying to organize or take part in a discussion that you may not have instigated and has been on-going for a long period. In these cases, references will be made to previous emails and arguments. As more and more important work happens over email lists, it is crucial that the participants and organizers of the discussion can track relevant arguments and their outcomes. However, this is far more difficult than it needs be currently since no management software exists that tracks emails at the level of argumentation. In this paper, we propose an argumentation ontology that seeks to make the first step in solving these problems. For our example domain, we focus on W3C (World Wide Web Consortium, a Web standards body) mailing lists.

In particular, we look at the goal-driven and semi-structured debates of the W3C Technical Architecture Group (TAG)¹ list. This email list is used by the TAG to accept, discuss, and resolve new issues. By “issue” we refer to a point of contention that has been formally adopted by the TAG or a W3C Working Group. To do this, the group takes an internal consensus to decide whether or not the issue falls within their scope of activities and has not previously been answered by other groups. Then the group, with input allowed from others through the list-serv, debates the issue until consensus on a resolution by the group has been found. While this issue-based system is structured, the email list is public and so contains many discussions among a potentially large pool of people.

* Authors’ names appear in alphabetical order.

¹ More information is available at <http://www.w3.org/2001/tag/>

The TAG currently use an issue tracking tool, the Extensible Issue Tracking System (EXIT) to record their own decisions.² However, this tool does not keep track of how the discussions on the listservs are connected to decisions in a principled manner. In trying to represent this structure of justification for particular actions, we have developed **Argontonaut**, an argumentation ontology (still in its early stages of development) that has been developed for the list-serv domain. This ontology may be used for other domains, and we discuss its extensibility and future directions of this research.

2 Related Work

Argumentation is commonly construed as involving a way of organizing statements in a way that allows for a structured justification of a particular position, where statements are seen as essentially defeasible and perhaps also subjective in some way. The representation of argument thus tries to expose how these justifications are linked. Unlike proof arguments, an agent is not committed to inferring a conclusion from a series of premises. But like proof arguments, one can see the structure through which a particular conclusion may be obtained. The modern study of argument was revived by Toulmin (1958) who investigated what is involved in establishing conclusions in the production of arguments, and modern argumentation schemes essentially derive from his attempts to represent this process.

The development of *Issue Based Information Systems* (IBIS) springs from work in the field of Computer Supported Collaborative Work (CSCW) by Rittel and Webber (1973). IBIS is composed of three distinct entities, *Issues*, *Positions* and *Arguments* consisting of the relations *supports*, *objects-to*, *replaces*, etc. (Buckingham Shum, 2003). An *issue* may be characterised in terms of a *position* that an agent may take, and an *argument* an agent may put forward for taking this *position*. In this way, *issues* may evolve in a manner which exposes their continual reformulation depending on the context of use.

It is common to look at the process of argumentation as being characterised by two distinct aspects: *having arguments*, which is where individuals collaboratively try to pursue disagreement and controversy, and *making arguments*, which is the process of giving justifications and making inferences from this to come to some form of action (O’Keefe, 1977; de Moorl and Efimova, 2004). We focus on the latter of these, and view the process of argumentation as being in some sense goal directed, where agents are working collaboratively to reach some shared objective by giving justifications for what it is they assert. Theorists also tend to distinguish different types of argument dialogues into *persuasive* dialogues involving conflicting points of view, *negotiation*, involving a conflict of interest and a need for cooperation and *deliberation*, where there is a need to agree on a course of action (Walton and Krabbe, 1995). In the domain of W3C mailing lists, the type of dialogue we encounter predominantly involves *deliberation*.

² More information available at <http://www.w3.org/2003/12/exit/>

On the level of argumentation moves or points, there are a plethora of different taxonomies to categorise different domains' needs. Pragmaticians and cognitive scientists talk about *coherence relations*, discourse analysts address *discourse relations* and those interested in communicative purpose talk about *dialogue acts* or *speech acts*. Seminal work by Mann and Thompson (1988) proposed *Rhetorical Structure Theory* (RST), a theory of text organisation, in which text spans (at the clausal level) are identified as being either a nucleus or a satellite, and relations are posited linking satellites to nuclei according to definitions of the rhetorical purpose of the text spans. RST posits *informational* and *presentational* rhetorical relations, where the former relate the content of the text spans themselves, while the latter define relations in terms of the effect the text spans have on the reader, so for example, *concession* is *presentational* while *contrast* is *informational*. DAMSL, a dialogue annotation approach proposed by Allen and Core (1997), goes on to distinguish dialogue acts on several levels, most notably distinguishing between *forward-looking functions* relating the current dialogue utterance to some future segment, e.g., *information-request*, and *backward looking functions* like *agreement*.

Speech act theory is a form of conversational analysis which has been particularly influential in CSCW. Winograd and Flores implemented speech act theory in the *Coordinator* project (1986) to model email conversations. They propose a general structure of *conversation for action*, which models conversations via finite state machine transitions. The *Coordinator* project attempted to structure email by getting users to annotate the *illocutionary* force (i.e., conventional interpretation of communicative action; Levinson, 1983) of emails, e.g., *reject*. Our approach is quite similar to the *Coordinator* approach except that instead of using speech acts, we use a hybrid between the rhetorical and speech act approaches for argument.

We have decided to use as our basis ontology the Resource Description Framework, a W3C Semantic Web standard (Manola and Miller, 2004). By using RDF, we minimally constrain our ontology. RDF provides us with "triples" of subjects, predicates (which can be constrained by domains and ranges), and objects, as well as sub-classing. However, it is unable to provide us with negation, disjoint classes, and other common formalisms. Yet for a formalization of a realm as notoriously informal as email communication, adopting a minimal formalism such as RDF is a strength, not a weakness. Besides the obvious point that in the case of W3C list-servs, the W3C is more likely to use a W3C standard than another non-W3C standard, RDF provides the two technical advantages of extensibility and a lack of contradiction via negation. In RDF, every statement is composed purely of URIs (Uniform Resource Identifiers, as in <http://www.example.com>), with each vocabulary item having its own globally unique URI. Any collection of RDF statements can be thought as a graph, and these graphs can then be merged by the trivial process of matching URIs. Any purely RDF "ontology" can be extended to interoperate with any other RDF ontology automatically by simply using vocabulary items from the two ontologies in a statement together.

In this manner, our general purpose argumentation ontology can be expanded to easily fit a large amount of domain-specific categories, such as in the W3C process. We purposefully try to interoperate with two ontologies under development. Firstly, the **coWork** ontology describing the W3C process by Sandro Hawke, is still in its early stages and we re-use some of their more stable classes.³ The other, the SIOC ontology for social communication, is more well-developed but does not cover list-serv communication in the detail needed, although we explicitly reuse some of their classes.⁴ Secondly, RDF constrains us to using URIs. While individual emails are given URIs (in the case of the W3C, by the W3C's list-serv management software), the individual "points" or textual fragments within an email are not given URIs. So any given email is likely to contain multiple points on possibly different arguments. Therefore, the ability to "free-form" label emails without the use of logical constructs such as negation is actually useful.

Our ontology has been informed by the analysis of list-serv communication, the W3C process documents, and the advice of interested W3C members. It has not yet been evaluated by a large study. However, in response to the needs of this domain we have tried to use a sufficiently detailed number of constructs that cover W3C process while at the same time minimizing the number of constructs needed to be deployed by a user. It is also exceedingly unlikely users will use an annotation tool to annotate their emails on the the sentence or paragraph level. Therefore, it is critical than our formalism allow the individual URI of an email message to be annotated with many different points on differing arguments, even if they are "in contradiction" to each other. We are not modelling ideal communication, but the more messy communication on email lists. In this regard, the minimalist formal semantics of RDF is ideal, as well as a more "free-form" style of annotation. However, the creation of easy-to-use email annotation tools to help lift the cognitive load off the users would be of great use.

There has been relatively little work on argumentation ontologies for the Semantic Web. The *DILIGENT* project proposes an argumentation ontology for the Semantic Web to model engineering design decisions, and they advocate decentralised and evolving collaborative ontology development. The main concepts in their ontology are *issues*, *ideas* and *arguments*, which are represented as classes. *Ideas* refer to how concepts should be formally represented in the ontology and relate ontology change operations. *Ideas* respond to *issues*, and indicate how they should actually be implemented in the ontology. *Arguments* debate either a particular *idea* or *issue*. They exemplify domain experts proposing new *issues* to be introduced into the ontology which are then argued over and formalised through concrete *ideas*. While this approach indicates how a distributed group can achieve consensus in constructing a shared ontology, their ontology does not extend neatly to modelling argumentation in emails, where there is not necessarily (and most often not) the construction of a formal ontology as the end result. Instead, the end result, if any, is more open-ended and may be prose,

³ Hawke's ontology is available at: <http://www.w3.org/2002/05/cowork/>

⁴ Available at: <http://www.rdfs.org/sioc/>

action items, software, and so on. Given the difference in their goals and ours, it is perhaps not surprising that their ontology does not model these more general purpose email discussions we are trying to model.

3 Argumentation in RDF

This ontology is designed to capture not just argumentation that occurs in emails on the mailing list, but also reference to software, documents, meeting transcriptions, and so on. We use the word “ontology” since we have formalized our domain, although this word in the context of the Semantic Web usually refers to ontologies built using OWL, a richer language built on top of RDF (McGuinness and van Harmelen, 2006). All of the below are subclasses of the RDF standard *Resource* class. Classes are listed in bold, while their properties are listed in italics, along with the domain and range of each property as the first and second argument in parenthesis respectively. We will denote subclass relationships for classes by indentation. Since our main area of interest for this paper is the properties of argumentation, we will first go over the main classes and properties of **Message**. It is important to note that the heart of the argumentation arises via the judicious use of the correct subclasses and properties of the *Message* class. Every statement in our ontology is at some point grounded in an actual email message.

- **Message**: corresponds to an email used in an argument.
 - **NewInfo**: introduces new information, whether document, software, topic or issue, etc.
 - **RequestInfo**: asks specifically for information.
 - **StatePrinciple**: abstracts away from current discussion to state general principles.
 - **UseCase**: puts forward a concrete use of a technology, a “use-case.”
 - **ProceduralPoint**: brings up formal procedure, as for example when calling for an official consensus may be needed to make progress.
 - **Example**: uses exemplification to make a point.
 - **Topic**: a discussion that has not yet been formalized or raised as an issue.
 - * **NewTopic**: the first mention of a point for discussion.

A sample of fifty messages in our domain showed that messages (emails) tend to be between 1 and 7 paragraphs long, and average to between 3 and 4 paragraphs per email. This naturally means that the author makes several points in the body of one message. This is quite reasonable given the interaction mechanism of the email, which facilitates making all one’s points in one message and then posting it, rather than making separate points in separate emails all successively posted. For us, this means that emails should be annotated with multiple tags, since presumably multiple points will have been made in a message, resulting in a bundle of argumentation tags describing what was communicated in the message rather than just one or two tags that define the a “single point” of the

argument as is the tradition in argumentation. However we do not have the many diverse relations we would otherwise have if we were annotating points rather than emails, which significantly reduces both the annotation burden and the bundle size of annotation tags. One benefit of this is a simpler taxonomy which is more readable, since we expect 2 or 3 tags, maximum 5 per email, rather than the 10-20 tags we would otherwise expect if we were annotating on the point level. Given the goal of browsing a mailing list and interpreting argumentation made in the emails, it makes more sense to annotate emails rather than points.

Implementation-wise, emails have URIs, which makes referring to them much easier than to points, which would have to have URIs assigned or created. Worse, users would be forced to use a custom annotation software to identify their points, while simply labelling an email is much more simple. For example, some at the W3C have suggested that users would be most comfortable annotating their emails by including any annotation in the subject line of their response, such as “[Disagree] CURIE Proposal” signifying a disagreement with a particular proposal. We believe this is approximately the level of cognitive load that a user would actually be capable of performing. However, for this to work there needs to be a series of rules that fills in the “missing” URIs of predicates. In the case of “disagree” it would be enough to simply automatically annotate it to be in disagreement. However, for more detail it is also easy enough to use the simple N3 notation in line in the text, which whom many members of the W3C are already familiar.⁵ This technique is already being pioneered by the the Semantic MediaWiki, which uses double square brackets to enclose RDF statements (Volkel et. al., 2006).

The relationship *refersTo* is the “glue” that our argumentation ontology uses to connect messages to each other. It can be distinguished from the *cite* relationship, since the *cite* relationship refers to prose documents while *refersTo* connects the more informal messages. It is different from the standard “Reply-To” header since often messages refer to other messages not directly in the header, but further back in the argument chain. The several different ways in which *refersTo* works which are characterised by its sub-properties, which are given below. These properties, combined with the classes given above, give the ontology the power it needs to track arguments.

- *refersTo(Resource, Resource)* The root “referring-to” relationship.
- *agree(Message, Message)*: in which the current message agrees with a previous message.
 - * *supportingExample(Message, Example)*: A concrete example that supports the content of a message.
- *disagree(Message, Message)*: the current message disagrees with a previous email.
 - * *modifyPoint(Message, Message)*: a reworking or changing of one aspect of a previous message.
 - * *counterExample(Message, Example)*: An example that contradicts the content of a previous message.

⁵ Given in detail at <http://www.w3.org/2000/10/swap/Primer>.

- *provideInfo(requestInfo, NewInfo)*: shows the connection between a request for information and an answer.

One possible problem with our ontology is that it ignores the fact that arguments by nature are dynamic, i.e. they evolve, as was argued above in the context of IBIS. The initial argument stated in the position email of the thread is rarely the same argument by the time two or three people have responded to it. People by nature tend to reinterpret and qualify arguments much like in the popular Chinese Whisper children's game, in which what is whispered from one person to the next is never the same thing at the end of the circle as at the beginning. However, messages posted to the W3C mailing lists tend to stay very much on topic, and qualifications rarely diverge from the topic. Also, since all of our argumentation is grounded in concrete messages rather than abstract points, to track the dynamics of the argument a user of the ontology merely has to refer to the text of the message(s), which is easily accessible via the message URI used by the ontology.

Agree and *Disagree* are usually thought to be mutually exclusive. This means that they can't both apply to the same message. Yet this type of relationship can only be expressed in the richer OWL language and cannot be expressed in RDF. However, this inability to formalize disjoint relationships is actually useful. We know that it's certainly reasonable for someone to play both the devil's advocate and advocate her own perspective regarding something under discussion in the same email. More frequently, the author could agree with one aspect of the preceding message while disagreeing with another aspect of the message's points.

4 W3C Process Ontology

While the argumentation ontology given in the previous section is broad enough to be used in other domains, a domain ontology detailing the W3C process is needed to demonstrate ontology-driven argumentation management. Wherever possible we have explicitly imported coWork classes in order to promote ontology re-use. While these classes are for the most part W3C-specific, general classes such as *Event* are imported from other ontologies and so can be interoperable with RDF Calendar and Semantic Web-based scheduling applications.⁶ The first group of classes has to deal with the issue-tracking process of the W3C, and how it interacts with the email list-servs and meetings.

- **Issue**: addresses topics that have been formally taken aboard the decision-making process of the group. Within the domain of the W3C they may use subclasses of *Issue* that distinguish them according to the stage of group acceptance which applies to them.
 - * **RaisedIssue**: when an issue is raised formally before a group.
 - * **AcceptedIssue**: when an issue is brought before a group and they agree to come to a decision on the issue.

⁶ RDF Calendar is a work in development available at www.w3.org/2002/12/cal/

- * **DeclinedIssue:** when an issue is brought before a groupe and they decide the issue is beyond their scope.
- * **AnnouncedDecision:** the official announcement of a decision on an issue.
- * **AgreedDecision:** an agreement, either through voting or consensus, of a decision.
- * **ProposedDecision:** the formal announcement of a proposed decision, the beginning of a formal feedback cycle.
- * **ObjectionToDecision:** a formal objection to a decision.
- **Event:** corresponds to a particular event, and is therefore *owl:sameAs* (<http://www.w3.org/2002/12/cal#event>), the event class from RDF Calendar.
 - * **Meeting** is a sub-class of Event and imported from *coWork*.
 - **DistributedMeeting** An IRC, video-conference, or other remote meeting.
 - **FacetoFaceMeeting** A face-to-face meeting.
 - *Consensus (Meeting, Issue)* : there was consensus over a decision regarding an issue.
 - *Dissent (Meeting, Issue)*: there was a dissent over a decision regarding an issue.
 - *Scribe (Meeting, Person)*: who took notes during the meeting. Should have a *refersTo* relationship with a **Notes** class.

The second part of the W3C Process ontology models in detail the development of a standard that addresses. The full formal process is given at <http://www.w3.org/2005/10/Process-20051014/tr>.

- **Document:**
 - * *addresses (Document, Issue)*: a document addresses a certain issue.
 - * **NonW3CStandardDocument:** a document not on the W3C Standard Track.
 - **Draft, Agenda, TeamSubmission, MemberSubmission,**
 - **Note**
 - * **PreW3CStandardDocument**
 - **DraftFinding:** a TAG finding.
 - **ApprovedFinding:** a TAG finding that has reached consensus.
 - **WorkingDraft:** the first stage of the W3C document process.
 - **CandidateRecommendation:** when the W3C feels a document fulfills the use cases and requirements needed for a standard, and now is in need of implementation.
 - **ProposedRecommendation:** when a standard is sent to the Advisory Committee for full review.
 - * **W3CStandardDocument**
 - **Recommendation:** when a specification has after extensive consensus-building received the endorsement of the W3C.
 - **RescindedRecommendation:** When due to some problem, the W3C has withdrawn a recommendation.

- * *authorOf (Person, Document)*: author of a document
- * *introduce (Message, Document)*: first email that introduces a document.
- * *cite (Message, Document)*: email that cites a document.
- * *requestComments (Message, Document)*: request for comments on a document.
- * *requestRevision (Message, Document)*: request for a textual revision of the document.
- * *introduceRevision (Message, Document)*: a textual revision has been made to a document.
- * *publicationDraft (Message, WorkingDraft)*, *lastCall (Message, WorkingDraft)*, *implementationsCall (Message, CandidateRecommendation)*, *reviewCall (Message, ProposedRecommendation)*, *recommendationPublication (Message, Recommendation)*: *rescindPublication (Message, Recommendation)*: These steps correspond to the various email announcements of the various promotions in a W3C standard-track document's lifecycle.
- **Software**: within the W3C domain we note whether or not software is an *implementationOf* of a particular W3C Standard, since standards require at least two implementations.
 - * **Beta**: not ready yet for release.
 - * **Release**: released.
 - * *implementationOf (Software, W3CStandardDocument)*: an implementation of a W3C Standard.
- **Group**: a group of people, the same as **Group** in the coWork Ontology. Groups may use the URI of their homepage, or create another one. The W3C has the following groups and relations, which are explained in W3C Process:
 - * **InterestGroup**, **WorkingGroup**, **CoordinationGroup**, **TechnicalArchitectureGroup**.
 - * *member(Person, Group)*, *chair(Person, Group)*, *teamContact(Person, Group)*, *InvitedExpert (Person, Group)*: an invited expert of a Group.

4.1 Some Examples

We will now turn to some examples, as given by excerpted text from emails in the W3C TAG list-serv, to illustrate how the ontology can be put to use. This email exchange brings up a possible concern about how a new standard has contradicted W3C policies as regards “namespaces,” which is the use of URIs to scope names in XML documents. These namespaces are often thought of as having a finite set of “names” in them. In particular, one member of the W3C is concerned that when one adds a new name to a namespace, like adding *xml:id* to the XML namespace, one should change the URI. These examples show how the N3 notation can be inserted in-line in text, using the double-bracket convention to annotate email text. Ideally a series of rules would let most annotation take place in the subject-line using single words, such as “[newTopic] Namespace Issues,” but these rules are currently still under development and beyond the scope of this paper.

Email:1

<http://lists.w3.org/Archives/Public/www-tag/2005Feb/0017.html>

Some of you may be aware that an issue with the
xml:id specification began erupting the day
before it became a CR.

[[email:1 a NewTopic.]]

[[email:1 cite <http://www.w3.org/TR/xml-id/>.]]

The kernel of the issue is my interpretation of
the definition of namespace as it appears in the
Namespaces recommendation. The definition is that
a namespace is a collection of names *identified*
by a URI.

So, for example, the namespace ({lang, space},
<http://www.w3.org/XML/1998/namespace>) is not equal to
({lang, space, base, id}, <http://www.w3.org/XML/1998/namespace>)
[[email:1 a Example.]]

The W3C director directed the W3C to this interpretation in
<http://www.w3.org/1999/10/nsuri>, which states that a recommendation
cannot add more than clarifications and bug fixes without changing
the namespace URI.

[[email:1 cite <http://www.w3.org/1999/10/nsuri>.
<http://www.w3.org/1999/10/nsuri> a Document.]]

First, this email introduces a *NewTopic* as a concern about how the particular *xml:id* standard was introduced. This email's concern has not officially been made into an issue. While there is a temptation to label this email with *newInfo*, we refrain from doing so. A *newInfo* label should be used in response to a specific question, as given by a *requestInfo*, and then connected by a *provideInfo* predicate. In the following examples we use “email ” to stand in for the URI of the actual email.

The label of *Example* above shows that the email contains a concrete example, and notice that the URI of the email is simply given the class of *Example* to denote this. All arguments to properties have URIs as was the case for the two usages of *cite* above, and these URIs are often given in-line in the text message as the URI of the supporting document. As shown by the reference to the *xml:id* by a verbal name and not a URI, one advantage of our argumentation system is that it explicitly reminds users to give URIs for cited documents. In summary, the combination of the two *cite* labels and the fact that it's a *newTopic* makes explicit within our ontology the possible contradiction between the example and the W3C policy as given by the cited document.

The email below is the response of a W3C TAG member to the concerns raised in Email 1. In order to save space we have deleted the “follow-up” citation of the previous email. Notice that the W3C TAG member is trying to clarify that new names can be added to a namespace without causing a change in the namespace

URI since this may not count as an “update” in the sense meant by the original document:

Email:2

<http://lists.w3.org/Archives/Public/www-tag/2005Feb/0022.html>
Which would be equivalent to saying that the state of a resource cannot change without also changing its URI. We know that is false in the general case,
[[email:2 a StatePrinciple.]]

so I hope you forgive me for rejecting **identified** as having the meaning you suggest.
[[email:2 disagree email:1.]]

No, it says that groups **should** use a template in the original namespace-defining document that specifies either that no updates will be made or that updates will be made only for clarification or bug fixes. It does not say whether adding a new name to the namespace is considered an update, nor whether such an update shall be considered major or minor.
[[email:2 cite <http://www.w3.org/1999/10/nsuri.>]]

Once an email has been annotated with a given relation, the author does not need to add multiple annotations for each following email that cites the previous email, since the granularity of annotation is on the level of the email and she/he has already annotated that email. Note that the follow-up email can use *disagree* to automatically refer a previous email, and then re-cite the same document with a different interpretation. Another email in the exchange is given below. This email does not explicitly disagree, but cites two documents that he believes covers the issue.

Email:3

<http://lists.w3.org/Archives/Public/www-tag/2005Feb/0024.html>
I think that the general issue of namespace evolution is currently covered in the extensibility and versioning finding. There is a section on component version identification choices which lists 4 broad based choices for ns management. Further, the web arch v1 says that specs should specify their namespace management policies.
[[email:3 cite <http://www.w3.org/2001/tag/doc/versioning.html>.
<http://www.w3.org/2001/tag/doc/versioning.html> a DraftFinding.
email:3 cite <http://www.w3.org/TR/webarch/>.
<http://www.w3.org/TR/webarch/> a Recommendation.]]

If you don't accept that the general part of your request is covered by finding/web arch, then can you elaborate on what is missing?
[[email:1 a requestInfo.]]

The author of the final email ends with a request for more information. Notice that if one was attempting to re-work and inspect standard documents, the ability to track the request for information that have remained unfulfilled would be useful.

5 Conclusions and Future Work

In this paper we have presented an argumentation ontology, with a particular focus on emails in the W3C mailing list domain. As is usually the case, the validation of utility arises out of use; that is, the evaluation of people using the ontology to annotate their emails will be of utmost importance. We hope that users of W3C mailing lists, which are often paid to participate and are at least committed to achieving the goals of the W3C and the Semantic Web, will take the needed time to annotate their emails according to our ontology.

The next area for future work is to deploy this ontology with a W3C Working Group in order to see if users would actually annotate their emails, most likely using by inserting the annotations in the subject line. If users find annotating their emails easy enough to accomplish, implementing a server-side tool which enables annotation of one's past emails as well as querying the emails using SPARQL would be obvious next step (Prud'hommeaux and Seaborne, 2006). We hope to explore how much this ontology scales to annotating argumentation in other domains.

References

1. Allen, J., Core, M.: Draft of DAMSL, Dialogue Annotation Mark-up in Several Layers (1997). www.cs.rochester.edu/research/speech/damsl/RevisedManual/.
2. Buckingham Shum, S. The Roots of Computer Supported Argument Visualization. In *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making*, P. Kirschner, S. Buckingham Shum and C. Carr (Eds.), Springer-Verlag, London (2003).
3. Levinson, S. *Pragmatics*. Cambridge University Press (1983).
4. Mann, W., Thompson, S.: *Rhetorical Structure Theory*. Text, vol.8 (1988).
5. Mc Guinness, D., van Harmelen, F. (eds.): *OWL Web Ontology Overview*. W3C Recommendation (2004), <http://www.w3.org/TR/owl-features/>.
6. de Moorl, A., Efimova, L.: An Argumentation Analysis of Weblog Conversations. *Proceedings of the 9th Int'l Conference on Perspective on Communication Modelling* (2004).
7. O'Keefe, D.: Two Concepts of Argument. *Journal of the American Forensic Association*, vol.13 (1977).
8. Prud'hommeaux, E., Seaborne, A. (eds.): *SPARQL Query Language*. W3C Candidate Recommendation (2006), <http://www.w3.org/TR/rdf-sparql-query/>.
9. Manola, F., Miller, E. (eds.): *RDF Primer*. W3C Recommendation (2004), <http://www.w3.org/TR/rdf-primer/>.
10. Tempich, C., Pinto, H.S., Sure, Y., Staab, S.: An Argumentation Ontology for Distributed, Loosely-controlled and evolving Engineering processes of ontologies. In *Second European Semantic Web Conference (ESWC 2005)*, 2005.
11. Toulmin, S.: *The Uses of Argument*. Cambridge University Press (1958).
12. Vlk M., Krtzsch M., Vrandecic D., Haller H., and Studer R. Semantic Wikipedia. In *Proceedings of the 15th Int'l Conference on World Wide Web (WWW 2006)*.
13. Walton, D., Krabbe, E.: *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. Albany, NY: SUNY Press (1995).
14. Winograd, T., Flores, F.: *Understanding Computers and Cognition: A New Foundation for Design*. Pearson Education, NJ (1986).

Clustering Approach Using Belief Function Theory

Sarra Ben Hariz, Zied Elouedi, and Khaled Mellouli

LARODEC, Institut Supérieur de Gestion de Tunis, 41 Avenue de la Liberté,
2000 Le Bardo, Tunisie

sarra.benhariz@gmail.com, zied.elouedi@gmx.fr,
khaled.mellouli@ihec.rnu.tn

Abstract. Clustering techniques are considered as efficient tools for partitioning data sets in order to get homogeneous clusters of objects. However, the reality is connected to uncertainty by nature, and these standard algorithms of clustering do not deal with this uncertainty pervaded in their parameters. In this paper we develop a clustering method in an uncertain context based on the K-modes method and the belief function theory. This so-called belief K-modes method (BKM) provides a new clustering technique handling uncertainty in the attribute values of objects in both the clusters' construction task and the classification one.

Keywords: machine learning, clustering, K-modes method, uncertainty, belief function theory.

1 Introduction

Clustering techniques [9] are among the well known machine learning techniques, and the K-modes method [8] is considered as one of the most popular of them. These techniques are used in many domains such as medicine, banking, finance, marketing, security, etc. They work under an unsupervised mode when the class label of each object in the training set is not known a priori. In addition to these unsupervised classification techniques, there exist those working under a supervised mode when the classes of instances are known in advance helping to construct a model that will be used for classifying new objects. Among them, we mention decision trees [12], k-nearest neighbor [3], neural networks [15], etc.

The capability to deal with datasets containing uncertain attributes is undoubtedly important due to the fact of this kind of datasets is common in real life data mining applications. However, this problem makes most of the standard methods inappropriate for clustering such training objects. In order to overcome this drawback, the idea is to combine clustering methods with theories managing uncertainty such as the belief function theory. This latter theory as interpreted in the transferable belief model (TBM) [16] presents an effective tool to deal with this uncertainty. It permits to handle partial or even total ignorance concerning classification parameters, and offers interesting means to combine several pieces of evidence. In fact, there are belief classification techniques which have been

developed such as belief decision trees (BDT) [5] and belief k-nearest neighbor [4], and which have provided interesting results.

The objective of this paper is to develop a new clustering method in an uncertain context that uses the K-modes paradigm and based on the belief function theory, the proposed approach is called the belief K-modes method (BKM). The main contributions of this paper are to provide one approach to deal with on one hand the construction of clusters where the values of the attributes of training objects may be uncertain, and in the other hand the classification of new instances characterized also by uncertain values based on the obtained clusters.

The remainder of this paper is organized as follows: Section 2 focuses on the basics of the K-modes method. Section 3 gives an overview of the belief function theory. Then, Section 4 presents our belief K-modes method where the two BKM parameters are detailed. Finally, our approach is illustrated by an example, and the experimental results are then presented and analyzed.

2 The K-Modes Method

The K-modes algorithm [8] was proposed to extend the K-means one [10] to tackle the problem of clustering large categorical data sets in data mining. This method uses a simple matching dissimilarity measure, modes instead of means for clusters, and a frequency-based approach to update modes in the clustering process to minimize the clustering cost function. The mentioned modifications to the K-means algorithm are discussed as follows: Let X_1, X_2 be two objects described by s categorical attributes. The dissimilarity measure between X_1 and X_2 can be defined by the total mismatches of the corresponding attribute categories of two objects. It can be defined as follows:

$$d(X_1, X_2) = \sum_{j=1}^s \delta(x_{1,j}, x_{2,j}), \text{ where : } \delta(x_{1,j}, x_{2,j}) = \begin{cases} 0 & \text{if } x_{1,j} = x_{2,j} \\ 1 & \text{if } x_{1,j} \neq x_{2,j} \end{cases}$$

Giving a cluster $C = \{X_1, \dots, X_p\}$ of p objects, with $X_i = (x_{i,1}, \dots, x_{i,s})$, $1 \leq i \leq p$, its mode $Q = (q_1, \dots, q_s)$ is defined by assigning q_j , $1 \leq j \leq s$, the category most frequently encountered in $\{x_{1,j}, \dots, x_{p,j}\}$. When the above is used as the dissimilarity measure for objects, the optimization problem for partitioning a set of n objects described by s categorical attributes into K clusters becomes:

$$\text{Minimize } P(W, Q) = \sum_{l=1}^k \sum_{i=1}^n \sum_{j=1}^s w_{i,l} \delta(x_{i,j}, q_{l,j}) \quad (1)$$

subject to: $\sum_{l=1}^k w_{i,l} = 1$, $1 \leq i \leq n$, $1 \leq l \leq k$, and $w_{i,l} \in \{0, 1\}$. where W is an $n \times k$ partition matrix, $w_{i,l}$ is the degree of membership of the object X_i in the cluster Q_l (using 1 and 0 to represent either the object X_i is an element of the cluster Q_l or not), and $Q = \{Q_1, Q_2, \dots, Q_k\}$. To minimize the cost function, the K-modes algorithm uses the following procedure:

1. Select K initial modes, one for each cluster.
2. Allocate an object to the cluster whose mode is the nearest to it according to the simple matching dissimilarity measure. Then, update the mode of the cluster after each allocation.
3. Once all objects have been allocated to clusters, retest the dissimilarity of objects against the current modes. If an object is found such that its nearest mode belongs to another cluster rather than its current one, then reallocate the object to that cluster and update the modes of both clusters.
4. Repeat step 3 until no object has changed clusters after a full cycle test of the whole data set.

Once the clusters are fixed, we have to classify a new instance based on its distance against the obtained clusters using the simple matching dissimilarity measure and assign it to the most closer one.

Let us note that the K-modes algorithm is unstable due to non-uniqueness of the clusters' modes.

3 Review of the Belief Function Theory

In this section, the basic concepts of the belief function theory as understood in the transferable belief model (TBM) are recalled briefly (for more details see [13], [14], [16]).

3.1 Background

Let Θ be a finite non empty set of elementary events to a given problem, called the frame of discernment. Θ contains hypotheses about one problem domain. The set of all the subsets of Θ is referred by the power set of Θ , denoted by 2^Θ .

The impact of a piece of evidence on the different subsets of the frame of discernment Θ is represented by the so-called basic belief assignment (bba). The bba is a function denoted m that assigns a value in $[0,1]$ to every subset A of Θ , and it is defined as follows: $m : 2^\Theta \mapsto [0, 1]$ such that $\sum_{A \subseteq \Theta} m(A) = 1$.

Each quantity $m(A)$, named basic belief mass (bbm) is considered as the part of belief that supports the event A , and that, due to the lack of information, does not support any strict subsets.

The belief function bel expresses the total belief fully committed to the subset A of Θ without being also committed to \bar{A} . This function is defined as follows: $bel : 2^\Theta \mapsto [0, 1]$, $bel(A) = \sum_{\phi \neq B \subseteq \Theta} m(B)$, where ϕ is the empty set.

With the belief function theory, it is easy to express the state of total ignorance. This is done by the so-called vacuous belief function which is defined by [13]:

$$m(\Theta) = 1 \text{ and } m(A) = 0 \text{ for all } A \subseteq \Theta, A \neq \Theta \quad (2)$$

On the other hand, this theory permits also to express the state of total certainty via the certain belief function which is defined as follows:

$$m(A) = 1 \text{ and } m(B) = 0 \text{ for all } B \neq A \text{ and } B \subseteq \Theta \quad (3)$$

where A is a singleton event.

3.2 Combination

Let m_1 and m_2 be two bba's defined on the same frame of discernment Θ . These two bba's are collected by two 'distinct' pieces of evidence and induced from two experts (information sources).

The bba that quantifies the combined impact of these two pieces of evidence is obtained through the conjunctive rule of combination [14].

$$(m_1 \wedge m_2)(A) = \sum_{B, C \subseteq \Theta; B \cap C = A} m_1(B)m_2(C) \quad (4)$$

3.3 Decision Making

The TBM is based on a two level mental models:

- The credal level where beliefs are entertained and represented by belief functions.
- The pignistic level where beliefs are represented by probability functions called pignistic probabilities. These probabilities are used to make decisions.

When a decision must be made, beliefs held at the credal level induce a probability measure at the pignistic measure denoted $BetP$ [16]. The link between these two functions, namely belief and probability functions is achieved by the pignistic transformation defined as follows:

$$BetP(A) = \sum_{B \subseteq \Theta} \frac{|A \cap B|}{|B|} \frac{m(B)}{(1 - m(\phi))}, \text{ for all } A \in \Theta \quad (5)$$

4 Belief K-Modes Method

Despite its accuracy when precise and certain data are available, the standard K-modes algorithm shows serious limitations when dealing with uncertainty. However, uncertainty may appear in the values of attributes of instances belonging to the training set that will be used to ensure the construction of clusters, and also in the classification of new instances which may be characterized by uncertain attribute values. To overcome this limitation, we propose to develop what we call a belief K-modes method (BKM), a new clustering technique based on the K-modes method within the belief function framework. In this part of our paper, we present the notations that will be used in the rest of the paper. Next, we define the two major parameters of the belief K-modes method needed to ensure both the construction and the classification tasks, namely clusters' centers and the dissimilarity measure.

4.1 Notations

The following notations will be used in the following:

T: a given data set of objects.

X_i : an object or instance, $i = 1, \dots, n$

$A = \{A_1, \dots, A_s\}$: a set of s attributes.

Θ_j : the frame of discernment involving all the possible values of the attribute A_j related to the classification problem, $j=1, \dots, s$

D_j : the power set of the attribute $A_j \in A$, where $D_j = 2^{\Theta_j}$.

$x_{i,j}$: the value of the attribute A_j for the object X_i .

$m^{\Theta_j}\{X_i\}$: expresses the beliefs on the values of the attribute A_j corresponding to the object X_i .

$m_i(c_j)$: denoted the bba given to $c_j \subseteq \Theta_j$ relative to object X_i .

4.2 The BKM Parameters

As with standard K-modes method, building clusters within BKM needs the definition of its fundamental parameters, namely, cluster modes and the dissimilarity measure. These parameters must take into account the uncertainty encountered in the training set and that pervade the attribute values of training objects.

Cluster Mode

Due to the uncertainty and contrary to the traditional training set where it includes only certain instances, the structure of our training set will be represented via bba's respectively to each attribute relative to each object, this training set offers a more generalized framework than the traditional one. Two extreme cases should be noted, when one attribute is known with certainty, it will be represented by a certain belief function (see Equation 3), whereas when it is missing we will use the vacuous belief function (see Equation 2). Within this structure of training set, our belief K-modes cannot use the strategy used by the standard method which is the frequency-based method to update modes of clusters.

The idea is to apply the mean operator to this uncertain context since it permits combining bba's respectively to each attribute provided by all objects belonging one cluster.

Note that using the mean operator offers many advantages since it satisfies these properties namely the associativity, the commutativity and the idempotency. The latter property is the most important one in our case. When we have two or more objects belonging to one cluster which provide the same bbm's corresponding to any uncertain attribute, their cluster's mode should be characterized by these same bbm's (provided by its objects).

Using the mean operator will solve the non-uniqueness problem of modes encountered in the standard K-modes method.

Given a cluster $C = \{X_1, \dots, X_p\}$ of objects, with $X_i = (x_{i,1}, \dots, x_{i,s})$, $1 \leq i \leq p$. Then, the mode of C is defined by : $Q = (q_1, \dots, q_s)$, with:

$$q_j = \{(c_j, m_{c_j}) | c_j \in D_j\} \quad (6)$$

where m_{c_j} is the relative bba of attribute value c_j within C .

$$m_{c_j} = \frac{\sum_{i=1}^p m_i(c_j)}{|C|} \quad (7)$$

with $C = \{X_1, X_2, \dots, X_p\}$ and $|C|$ is the number of objects on C . m_{c_j} expresses the belief about the value of the attribute A_j corresponding to the cluster mode.

Dissimilarity Measure

The dissimilarity measure has to take into account the bba's for each attribute for all objects in the training set, and compute the distance between any object and each cluster mode (represented by bba's). Many distance measures between two bba's were developed which can be characterized into two kinds:

- Distance measures based on pignistic transformation [1], [6], [17], [18]: For these distances, one unavoidable step is the pignistic transformation of the bba's. Since, there is no bijection between bba's and pignistic probabilities (transformation from the power set to the set). This kind of distance may lose information given by the initial bba's. Besides, we can obtain the same pignistic probabilities by applying the pignistic transformation on two different bba's distributions. So, the distance between the two obtained results does not reflect the actual similarity between the starting bba's distributions
- Distances measures between bba's defined on the power set [2], [7]: The second one developed by Fixen and Mahler [7] is a pseudo-metric, since the condition of nondegeneracy of one distance metric is not respected.

Our idea is to adapt the belief distance defined by [2] to this uncertain clustering context to compute the dissimilarity between any object and each cluster mode. This distance measure takes into account both the bba's distributions provided by the objects and one similarity matrix D which is based on the cardinalities of the subsets of the corresponding frame of one attribute and those of the intersection and union of these subsets.

Let m_1 and m_2 be two bba's on the same frame of discernment Θ_j , the distance between m_1 and m_2 is :

$$d(m_1, m_2) = \sqrt{\frac{1}{2}(m_1^{\rightarrow} - m_2^{\rightarrow})D(m_1^{\rightarrow} - m_2^{\rightarrow})} \quad (8)$$

Another way to write it is:

$$d(m_1, m_2) = \sqrt{\frac{1}{2}(\|m_1^{\rightarrow}\|^2 + \|m_2^{\rightarrow}\|^2 - 2 \langle m_1^{\rightarrow}, m_2^{\rightarrow} \rangle)} \quad (9)$$

where $\langle m_1^{\rightarrow}, m_2^{\rightarrow} \rangle$ is the scalar product defined by:

$$\langle m_1^{\rightarrow}, m_2^{\rightarrow} \rangle = \sum_{w=1}^{2^{\Theta_j}} \sum_{z=1}^{2^{\Theta_j}} m_1(B_w) m_2(B_z) \frac{|B_w \cap B_z|}{|B_w \cup B_z|} \quad (10)$$

with $B_w, B_z \in D_j$ for $w, z = 1, \dots, 2^{\Theta_j}$, and $\|\mathbf{m}^{\rightarrow}\|^2$ is then the square norm of $\mathbf{m} : \|\mathbf{m}^{\rightarrow}\|^2 = \langle \mathbf{m}^{\rightarrow}, \mathbf{m}^{\rightarrow} \rangle$

This scalar product is based on the bba's distributions (m_1 , and m_2) and the elements of one similarity matrix D , which are defined as follows:

$$D(B_w, B_z) = \frac{|B_w \cap B_z|}{|B_w \cup B_z|}, \text{ where } B_w, B_z \in D_j.$$

Thus, the dissimilarity measure between any object X_i and each mode Q can be defined as follows:

$$D(X_i, Q) = \sum_{j=1}^m d(m^{\Theta_j}\{X_i\}, m^{\Theta_j}\{Q\}) \quad (11)$$

where $m^{\Theta_j}\{X_i\}$ and $m^{\Theta_j}\{Q\}$ are the relative bba of the attribute A_j provided by respectively the object X_i and the mode Q .

4.3 The BKM Algorithm

The BKM algorithm has the same skeleton as standard K-modes method. The different construction steps of our approach are described as follows:

1. Giving K , the number of clusters to form.
2. Partition objects in K nonempty subsets.
3. Compute seed points as the clusters' modes of the current partition using the mean (see Equation 7).
4. Assign each object to the cluster with the nearest seed point after computing the distance measures defined in Equation 11.
5. Go back to step 4, stop when no more new assignment.

Once the clusters' construction is done, the classification of a new object that may be characterized by uncertain attribute values, we have to assign it to the most similar cluster based on its distance over the obtained clusters resulting from the construction phase, and using the distance measure (See Equation 11).

Example 1. Let us illustrate our method by a simple example. Assume that a firm wants to group its staff by taking into account a number of their attributes. Let T be a training set composed of seven instances characterized by three categorical attributes: Qualification with possible values $\{A, B, C\}$., Income with possible values $\{High (H), Low (L), Average (Av)\}$., and Department with possible values $\{Finance (F), Accounts (Ac), Marketing (M)\}$. For each attribute A_j for an object X_i belonging to the training set T , we assign a bba $m^{\Theta_j}\{X_i\}$ expressing beliefs on its assigned attributes values, defined respectively on $\Theta_1 = \{A, B, C\}$, $\Theta_2 = \{H, L, Av\}$, $\Theta_3 = \{F, Ac, M\}$. If we consider that only the department attribute is known with uncertainty. The structure of the data set T can be defined in Table 1.

Let us now try to construct the clusters using our approach relative to the training set T . The first step is to specify K the number of clusters to form, and select the initial modes. Suppose that $K = 2$, 2-partition of T is initialized randomly as follows: $C_1 = \{X_1\}$, and $C_2 = \{X_2\}$.

We have to compute the distance measures relative to all objects $\{X_3, X_4, X_5, X_6, X_7\}$ and the 2 initial modes. For example, for the object X_3 , and after computing its distance measure over the two clusters' modes, we

Table 1. Data set T relative to BKM

Objects	Qualification	Income	Department
X_1	A	H	$m_1(F) = 0.5$ $m_1(\{F, A\}) = 0.3$ $m_1(\Theta_3) = 0.2$
X_2	B	L	$m_2(F) = 0.8$ $m_2(\Theta_3) = 0.2$
X_3	C	Av	$m_3(M) = 0.8$ $m_3(\{F, A\}) = 0.1$ $m_3(\Theta_3) = 0.1$
X_4	C	Av	$m_4(Ac) = 0.8$ $m_4(\Theta_3) = 0.2$
X_5	B	L	$m_5(M) = 0.8$ $m_5(\Theta_3) = 0.2$
X_6	A	H	$m_6(\{F, A\}) = 0.8$ $m_6(\Theta_3) = 0.2$
X_7	B	L	$m_7(A) = 0.8$ $m_7(\Theta_3) = 0.2$

obtain $d(X_3, Q_1) = 2.723$ and $d(X_3, Q_2) = 2.776$, so X_3 is assigned to C_1 since $d(X_3, Q_1) < d(X_3, Q_2)$. It is the same for the fourth other objects. After that all objects have been assigned to appropriate clusters, the following clusters are obtained $C_1 = \{X_1, X_3, X_6\}$, and $C_2 = \{X_2, X_5, X_4, X_7\}$. Next, we have to update clusters' modes. The same steps will be applied until no object has changed clusters. We finally obtain these clusters: $C_1 = \{X_1, X_3, X_4, X_6\}$, and $C_2 = \{X_2, X_5, X_7\}$, with the corresponding modes : $Q_1 = (\{(A, 0.5), (C, 0.5)\}; \{(H, 0.5), (Av, 0.5)\}; \{(F, 0.125), (Ac, 0.2), (M, 0.2), (\{F, A\}), 0.2), (\{F, A, M\}), 0.275\})$, and $Q_2 = (\{(B, 1)\}, \{(L, 1)\}; \{(F, 0.267), (Ac, 0.267), (M, 0.266), (\{F, A, M\}), 0.2\})$ Once, the two clusters are fixed, suppose that we would classify a new object X_i characterized by certain and exact values for its qualification and income attributes which are respectively the values B and Low. However, there is some uncertainty in the value of the department attribute defined by: $m_i(F) = 0.4$; $m_i(\{F, M\}) = 0.3$; $m_i(\Theta_3) = 0.3$. As a result, we obtain that the new instance to classify has respectively 1.355 and 0.300 as distances respectively the two clusters. So, this object is assigned to the second cluster C_2 since $d(X_i, Q_2) < d(X_i, Q_1)$.

5 Experimental Results

For the evaluation of the proposed BKM method, we have developed programs in Matlab V6.5, which consist in both building and the classification procedures corresponding to our approach. Then, we have applied these programs to real databases obtained from the U.C.I repository of Machine Learning databases [11]. We have modified these databases by introducing uncertainty in the attributes' values of their instances. A brief description of these databases is presented in Table 2.

Huang [8] proposed a measure of clustering results called the clustering accuracy r computed as follows: $r = \frac{\sum_{i=1}^k a_i}{n}$, where n is the number of instances in the dataset, k is the number of clusters, a_i is the number of instances occurring in both cluster i and its corresponding labeled class. This criterion is equivalent to the PCC expressing the percentage of the correctly classified instances. In order to evaluate the BKM approach, for each data set, we run the algorithm several times. The accuracy of our results is measured according to the mean PCC crite-

Table 2. Description of databases

Database	#instances	#attributes	#classes
Congressional voting records database	497	16	2
Balance scale database	625	4	3
Wisconsin breast cancer database	690	8	2

Table 3. Experimental results

Database	PCC
Congressional voting records database	86.52
Balance scale database	79.20
Wisconsin breast cancer database	71.39

tion of the obtained ones. Table 3 summarizes different results relative to these three databases obtaining by applying our proposed BKM approach.

It is found that the clustering results produced by the proposed method are very high in accuracy. The PCC's show that our method presented interesting results. So, These results confirm that our approach is well appropriate within the uncertain context. We should mention that with our proposed approach, and if all the bba's are certain then the obtained results as equivalent to ones corresponding to the standard K-modes method.

6 Conclusion

In this paper, we have developed a new clustering approach using the K-modes paradigm to handle uncertainty within belief function framework. It allows us to construct clusters within objects having uncertain attributes' values. Another advantage of BKM method is that once the clusters are fixed, the classification of new instances that may be uncertain is possible. Our future work is to extend our belief K-modes method to cluster data sets with mixed numeric and categorical attributes.

References

1. M. Bauer, Approximations for efficient computation in the theory of evidence, *Artif.Intell.* 61(2), 315-329, 1993.
2. E. Bosse, A-L. Jousselme, and D. Grenier: A new distance between two bodies of evidence. In *Information Fusion 2*, 91-101, 2001.
3. T.M. Cover and P.E. Hart, Nearest neighbor pattern classification, *IEEE Trans Inform Theory*, vol. IT-13, 21-27, 1967.
4. T. Denoeux, A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man and Cybernetics*, 25 (5), 804-813, 1995.
5. Z. Elouedi, K. Mellouli, P. Smets. Belief Decision trees: Theoretical foundations. *International Journal of Approximat Reasoning*, Vol 28, Issues 2-3, 91-124, 2001.

6. Z. Elouedi, K. Mellouli, and Ph. Smets. Assessing sensor reliability for multisensor data fusion within the transferable belief model. *IEEE Trans Syst Man Cybern B Cybern*, 34(1), 782-787, 2004.
7. D. Fixen and R.P.S. Mahler, The modified Dempster-Shafer approach to classification, *IEEE Trans.Syst.Man Cybern. A* 27(1), 96-104, 1997.
8. Z. Huang: Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining Knowl.Discov.*, Vol.2, No.2, 283-304, 1998
9. A.K. Jain and R.C. Dubes: Algorithms for clustering data. Prentice-Hall, Englewood cliffs, NJ, 1988.
10. J. MacQueen: Some methods for classification and analysis of multivariate observations. In: *Proc.of the Fifth Berkeley Symposium on Math, Stat. and Prob* Vol 1, 281-296, 1967.
11. P.M. Murphy and D.W. Aha. Uci repository databases. <http://www.ics.uci.edu/mlearn>., 1996.
12. J.R. Quinlan, Learning efficient classification and their application to chess end games. In R. S. Michalski, J. G. Carbonell, and T. M. Michell (Eds.), *Machnie Learning: An artificial intelligence approach*, 463-482. Morgan Kaufmann, 1983.
13. G. Shafer A mathematical theory of evidence. Princeton Univ. Press. Princeton, NJ, 30, 1976.
14. Ph. Smets and R. Kennes: The transferable belief model. *Artificial Intelligence*, 66: 191-234, 1994.
15. D.E. Rumelhart, G.E. Hinton, and R.J. Williams, Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing*. Cambridge, MA: MIT Press, 1986.
16. Ph. Smets: The transferable belief model for quantified belief representation. In D. M. Gabbay and P. Smets (Eds.), *Handbook of defeasible reasoning and uncertainty management systems*, Vol 1, 267-301, 1998b.
17. B. Tessem, Approximation algorithms and decision making in the Dempster-Shafer theory of evidence - an empirical study, *Int.J.Approx.Reason.* 17(2-3) 217, 1997.
18. L.M. Zouhal and T. Denoeux, An evidence-theory k-NN rule with parameter optimization, *IEEE Trans.Syst.Man Cybern. C* 28(2), 263-271, 1998.

Machine Learning for Spoken Dialogue Management: An Experiment with Speech-Based Database Querying

Olivier Pietquin*

Supélec – Campus de Metz, rue Edouard Belin 2,
F-57070 Metz – France
olivier.pietquin@supelec.fr

Abstract. Although speech and language processing techniques achieved a relative maturity during the last decade, designing a spoken dialogue system is still a tailoring task because of the great variability of factors to take into account. Rapid design and reusability across tasks of previous work is made very difficult. For these reasons, machine learning methods applied to dialogue strategy optimization has become a leading subject of researches since the mid 90's. In this paper, we describe an experiment of reinforcement learning applied to the optimization of speech-based database querying. We will especially emphasize on the sensibility of the method relatively to the dialogue modeling parameters in the framework of the Markov decision processes, namely the state space and the reinforcement signal. The evolution of the design will be exposed as well as results obtained on a simple real application.

Keywords: Spoken Dialogue Systems, Reinforcement Learning, Dialogue Management.

1 Introduction

In the last few years, research in the field of Spoken Dialogue Systems (SDS) has experienced increasing growth. But, the design of an efficient SDS does not basically consist in combining speech and language processing systems such as Automatic Speech Recognition (ASR) and Text-to-Speech (TTS) synthesis systems. It requires the development of an interaction management strategy taking at least into account the performances of these subsystems (and others), the nature of the task (i.e. form filling or database querying) and the user's behavior (i.e. cooperativeness, expertise). The great variability of these factors makes rapid design of dialogue strategies and reusability across tasks of previous work very complex. For these reasons, machine learning techniques applied to strategy optimization is currently a leading domain of researches. When facing a learning problem, two main classes of methods could be envisioned: supervised and unsupervised learning. Yet, supervised learning would require examples of ideal (sub)strategies which are typically unknown. Indeed, no one can actually provide an example of what would have objectively been the perfect

* This work was realized when the author was with the Faculty of Engineering, Mons (FPMs, Belgium) and was sponsored by the 'Direction Générale des Technologies, de la Recherche et de l'Énergie' of the Walloon Region (Belgium, First Europe Convention n° 991/4351).

sequencing of exchanges after having participated to a dialogue. Humans have a greater propensity to criticize what is wrong than to provide positive proposals. In this context, Reinforcement Learning (RL) [1] appears as the best solution to the problem and have been first proposed in [2] and further developed in [3][4][5]. The main differences between the approaches rely in the way they model the dialogue manager's environment during the learning process. In [2] and [5], the environment is modeled as a set of independent modules (i.e. ASR system, user) processing information (this approach will be adopted in this paper). In [3], the environment is modeled as a pure state-transition system for which parameters are learned from dialogue corpora. In [4], a hybrid approach is described.

However, transposing spoken dialogue management in the formalism of the Markov Decision Processes (MDP) is required in every approach and it is probably the crucial point. No rational method but common sense is generally used to define the parameters of the corresponding MDP. In this paper, we emphasize on the importance of these parameters. The sensible steps are mainly the state space definition and the choice of the reinforcement signal. This is demonstrated on a simple speech-based database querying application.

2 Markov Decision Processes and Dialogue Management

2.1 MDP and Reinforcement Learning

In the MDP formalism, a system is described by a finite or infinite number of states $\{s_i\}$ in which a given number of actions $\{a_j\}$ can be performed. To each state-action pair is associated a transition probability \mathcal{T} giving the probability of stepping from state s at time t to state s' at time $t+1$ after having performed action a when in state s . To this transition is also associated a reinforcement signal (or reward) r_{t+1} describing how good was the result of action a when performed in state s . Formally, an MDP is thus completely defined by a 4-tuple $\{S, A, \mathcal{T}, \mathcal{R}\}$ where S is the state space, A is the action set, \mathcal{T} is a transition probability distribution over the state space and \mathcal{R} is the expected reward distribution. The couple $\{\mathcal{T}, \mathcal{R}\}$ defines the dynamics of the system:

$$\begin{aligned}\mathcal{T}_{ss'}^a &= \mathbf{P}(s_{t+1} = s' | s_t = s, a_t = a) \\ \mathcal{R}_{ss'}^a &= \mathbf{E}[r_{t+1} | s_t = s, a_t = a, s_{t+1} = s']\end{aligned}\tag{1}$$

These last expressions assume that the Markov property is met, which means that the system's functioning is fully defined by its one-step dynamics and that the functioning from state s will be identical whatever the path followed until s . To control a system described as an MDP (choosing actions to perform in each state), one would need a *strategy* or *policy* π mapping states to actions: $\pi(s) = \mathbf{P}(a|s)$ (or $\pi(s) = a$ if the strategy is deterministic).

In this framework, a RL *agent* is a system aiming at optimally mapping states to actions, that is finding the best strategy π^* so as to maximize an overall reward R which is a function (most often a weighted sum) of all the immediate rewards r_t . If the probabilities of equations (1) are known, an analytical solution can be computed by dynamic programming [1], otherwise the system has to learn the optimal strategy by a

trial-and-error process. RL is therefore about how to optimally map situations to actions by trying and observing *environment's* feedback. In the most challenging cases, actions may affect not only the immediate reward, but also the next situation and, through that, all subsequent rewards. Trial-and-error search and delayed rewards are the two main features of RL. Different techniques are described in the literature, in the following (mainly in section 4) the Watkin's $Q(\lambda)$ algorithm [1] will be used.

2.2 Dialogue Management as an MDP

As depicted on Fig.1, a task-oriented (or goal-directed) man-machine dialogue can be seen as a turn-taking process in which a human user and a Dialogue Manager (DM) exchange information through different channels processing speech inputs and outputs (ASR, TTS ...). In this application, the DM *strategy* has to be optimized and the DM will be the learning *agent*. Thus the *environment* modeled by the MDP comprises everything but the DM: the human user, the communication channels (ASR, TTS ...), and any external information source (database, sensors etc.). In this context, at each turn t the DM has to choose an *action* a_t according to its interaction *strategy* so as to complete the task it has been designed for. These actions can be greetings, spoken utterances (constraining questions, confirmations, relaxation, data presentation etc.), database queries, dialogue closure etc. They result in a response from the DM environment (user speech input, database records etc.), considered as an observation o_t , which usually leads to a DM *internal state* update. To fit to the MDP formalism, a *reinforcement signal* r_{t+1} is required. In [3] it is proposed to use the contribution of an action to the user's satisfaction. Although this seems very subjective, some studies have shown that such a reward could be approximated by a linear combination of objective measures such as the duration of the dialogue, the ASR performances or the task completion [6]. A practical example will be provided subsequently.

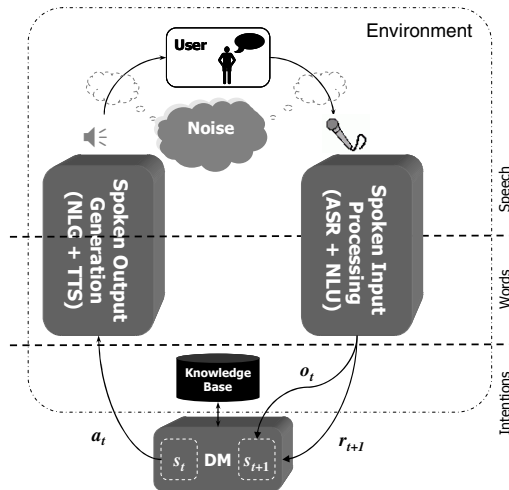


Fig. 1. Dialogue management as an MDP. NLG stands for Natural Language Generation and NLU stands for Natural Language Understanding

3 Speech-Based Database Querying

The considered task consists in a database querying system. The goal of the application is to present a list of computers selected in a database according to specific features provided by a user through speech-based interaction.

The database contains 350 computer configurations split into 2 tables (for notebooks and desktops), each of them containing 6 fields: `pc_mac` (pc or mac), `processor_type`, `processor_speed`, `ram_size`, `hdd_size` and `brand`.

3.1 Action Set

Since the task involves database querying, actions will not only imply interaction with the user (such as spoken questions, confirmation requests or assertions) but also with the database (such as database querying). The action set contains 8 generic actions:

- `greet`: greeting (e.g. “How may I help you ?”).
- `constQ(arg)`: ask to constrain the value of *arg*.
- `openQ`: ask an open ended question.
- `expC(arg)`: ask to confirm the value of *arg*.
- `allC`: ask for a confirmation of all the arguments.
- `rel(arg)`: ask to relax the value of *arg*.
- `dbQ([args])`: perform a database query thanks to retrieved information.
- `close`: present data and close the dialogue session.

The value of *arg* may be the table’s type (notebook or desktop) or one of the 6 table fields. Notice that there is not data presentation action because it will be considered that the data presentation is included in the ‘close’ action.

3.2 State Space

Building the state space is a very important step and several state spaces can be envisioned for the same task. Yet, some general considerations might be taken into account:

1. The state representation should contain enough information about the history of the dialogue so the Markov property can be assumed.
2. State spaces are often considered as informational in that sense that they are built thanks to the amount of information the DM could retrieve from the environment until it reached the current state.
3. The state representation must embed enough information so as to give an accurate representation of the situation to which an action has to be associated (it is not as obvious as it sounds).
4. The state space must be kept as small as possible since the RL algorithms converge in linear time with the number of states of the underlying MDP.

According to these considerations and the particular task of database querying, two slightly different state spaces were built to describe the task as an MDP to illustrate the

sensitivity of the method to the state space representation. In the first representation, referred to as S_1 in the following, each state is represented by two features.

- A vector of 7 boolean values $[f_x]$ (one for each value of arg). Each of these f_x is set to *true* if the corresponding value of arg is known (for example if the user specified to search in the notebooks table, f_o is set to *true*). This is a way to meet the Markov property (informational state).
- Information about the Confidence Level (CL) of each f_x set to *true*. The confidence level is usually a real number ranging between 0 and 1 computed by the speech and/or language analysis subsystems (ASR and NLU) and providing information about the confidence of the system in the result of its processing. To keep the size of the state space reasonable, we only considered 2 possible values for the confidence level: *High* or *Low* (i.e. *High* means $CL \geq 0.8$ and *Low* means $CL < 0.8$).

Notice that ‘dbQ’ actions will only include values with a *High* confidence level. For each value of arg , there are 3 different possibilities for the corresponding slot in the state representation: $\{f_x = false, CL = undef\}$, $\{f_x = true, CL = Low\}$, $\{f_x = true, CL = High\}$. This leads to 3^7 possible states.

The second state representation is built on the same basis but an additional state variable NDB is added to take the number of records returned by the last ‘dbQ’ action into account. This variable can also take only two values (*High* or *Low*) and is set according to the comparison of the query result size and a predefined threshold. If no ‘dbQ’ action has been performed, the NDB variable is initialized with the *High* value (an empty query would provide the whole database as a result). This state space representation will be referred to as S_2 in the following.

3.3 Reward Function

Again, several proposals can be made for building the reward function and slight differences in the choices can result in large variations in the learned strategy. To illustrate this, some simple functions will be described in the following. According to [6], the reward function (which is here a cost function that we will try to minimize) should rely on an estimate of the dialogue time duration (D), the ASR performances (ASR) and the task completion (TC) so as to approximate the user’s satisfaction using objective measures:

$$R = w_D \cdot D - w_{ASR} \cdot ASR - w_{TC} \cdot TC \quad (2)$$

In this last expression, the w_x factors are positive weights. Considering the estimate of the time duration, two values are actually available: the number of user turns $D = N_U$ (the number of turns perceived by the user) and the number of system turns $D = N_S$ (including database queries as well).

On another hand, the task completion is not always easy to define. The *kappa* coefficient defined in [6] is one possibility but didn’t always prove to correlate well with the perceived task completion. For the purpose of this experiment, two simple task completion measures will be defined:

$$TC_{\max} = \max(\#(G_U \cap R)) \quad (3)$$

$$TC_{av} = average(\#(G_U \cap R)) \quad (4)$$

In these last expressions $\#(G_U \cap R)$ is the number of common values in the user's goal G_U (the user goal is supposed to have the same structure as an existing database record and is set before the dialogue begins) and one of the records R presented to the user at the end of a dialogue. When a value is not present in the user goal it is considered as common (if a field is not important to the user, it is supposed to match any value). The first task completion measure TC_{max} indicates how close the closest record in the presented results is. The second TC_{av} measures the mean number of common values between the user's goal and each presented record.

Finally, the ASR performance measures will be provided by the confidence levels (CL) computed by the ASR system after each speech recognition task.

4 Experiments

The number of required interactions between a RL agent and its environment is quite large (10^4 dialogues at least in our case). So, it has been mandatory to simulate most of the dialogues for two main reasons. First, it is very difficult (and expensive) to obtain an annotated dialogue corpus of that size. Second, it would have been too time consuming to realize this amount of spoken dialogues for training. So, in a first approximation, a written-text-based simulation environment has been built [5]. It simulates ASR errors using a constant Word Error Rate (WER) and provides confidence levels according to a distribution measured on a real system. If the system has to recognize more than one argument at a time, the CL is the product of individual CL s obtained for each recognition task (so it decreases). Other ASR simulation models can be considered [7] but it is out of the scope of this paper.

Several experimental results obtained with different settings of the state space and the reward function will be exposed in the following. These settings are obtained by combining in three different ways the parameters S_I , S_2 , N_U , N_S , TC_{max} , TC_{av} mentioned before. Results are described in terms of average number of turns (user and system), average task completion measures (TC_{max} and TC_{av}) for the performance and in terms of action occurrence frequency during a dialogue session to get a clue about the learned strategy. These results are obtained by simulating 10,000 dialogues with the learned strategy.

4.1 First Experiment: S_I , N_U , TC_{max}

The first experiment is based on the smaller state space S_I (without any clue about the number of retrieved records). The dialogue cost is computed thanks to the number of user turns N_U as a measure of the time duration and the TC_{max} value as the task completion measure. Results are as follows:

Table 1. Performances of the learned strategy for the $\{S_I, N_U, TC_{max}\}$ configuration

N_U	N_S	TC_{max}	TC_{av}
2.25	3.35	6.7	1.2

Table 2. Learned strategy for the $\{S_I, N_U, TC_{max}\}$ configuration

greet	constQ	openQ	expC	AllC	rel	dbQ	close
1.00	0.06	0.0	0.14	0.0	0.05	1.10	1.00

When looking at the three first columns of the performance table, the learned strategy doesn't look so bad. It actually has a short duration in terms of user turns as well as in system turns and has a very high task completion rate in terms of TC_{max} measure. Yet the TC_{av} shows a very low mean value.

When looking to the average frequency of actions in table, one can see that the only action addressed to the user that happens frequently during a dialogue is the greeting action. Others almost never happen. Actually, the learned strategy consists in uttering the greeting prompt to which the user should answer by providing some argument values. Then the system performs a database query with the retrieved attributes and provides the results to the user. Sometimes, the user doesn't provide any attribute when answering to the greeting prompt or the value is not recognized at all by the ASR model, so the strategy is to perform a constraining question (and not an open ended question) that will provide an argument with a better CL . Sometimes the provided arguments have a poor CL and an explicit confirmation is asked for. Sometimes the provided arguments don't correspond to any valid record in the database so the strategy is to ask for relaxation of one argument (this also explains why the number of database queries is greater than 1). The value of TC_{max} is not maximal because sometimes the dialogue fails.

This results in presenting almost all the database records when the user only provides one argument when prompted by the greeting. This is why there is a so big difference between TC_{max} and TC_{av} . The desired record is actually in the presented data (TC_{max} is high) but is very difficult to find (TC_{av} is low). The learned strategy is definitely not suitable for a real system.

4.2 Second Experiment: S_2, N_U, TC_{av}

This experiment uses the same settings as the previous one except that the NDB variable is added to the state variables and the task completion is measured with TC_{av} . Results are as follows:

Table 3. Performances of the learned strategy for the $\{S_2, N_U, TC_{av}\}$ configuration

N_U	N_S	TC_{max}	TC_{av}
5.75	8.88	6.7	6.2

Table 4. Learned strategy for the $\{S_2, N_U, TC_{av}\}$ configuration

greet	constQ	openQ	expC	AllC	rel	dbQ	close
1.00	0.87	1.24	0.31	1.12	0.21	3.13	1.00

This time, TC_{max} and TC_{av} are close to each other, showing that the presented results are more accurate but the number of turns has increased. The number of system turns particularly shows higher values. This observation is obviously explained by the increase of database queries.

Looking at the action occurrence frequencies one can see that the learning agent tries to maximize the TC_{av} value while minimizing the number of user turns and maximizing recognition performance. To do so, it always performs a database query after having retrieved information from the user. Since the number of results is part of the state representation, the agent learned not to present the results when in a state with a high number of results. If this number is too high after the greeting, the learner tries to reach a state where it is lower. Thus it almost systematically performs an ‘openQ’ action after the greeting in order to get as much information as possible in a minimum of turns (this explains the 1.24 value). Yet, this often results in poorer recognition outputs, thus it also performs a confirmation of all the fields before presenting any result. Sometimes, more information is provided after the greeting and only a constraining question is needed to gather enough information to reach a state with less result. A constraining question is preferred in this case because it leads to better recognition results.

The mean number of user turns shows that only 5.75 turns are usually needed to reach an accurate result set because the computer configurations are sufficiently different so as not to need too much attributes in the database query to provided accurate results. Thus, the system doesn’t ask for all the attribute values to the user. Further investigations would show that the system takes advantage of the structure of the database and asks for attributes allowing extracting the desired records as fast as possible.

4.3 Third Experiment: S_2, N_S, TC_{av}

The same experiment as the previous one has been performed but replacing the N_U measure of time duration by the N_S measure. It actually makes sense since in a real application, the database could be much larger than the one used here. Thus, the database queries could be much more time consuming.

Table 5. Performances of the learned strategy for the $\{S_2, N_S, TC_{av}\}$ configuration

N_U	N_S	TC_{max}	TC_{av}
6.77	7.99	6.6	6.1

Table 6. Learned strategy for the $\{S_2, N_S, TC_{av}\}$ configuration

greet	constQ	openQ	expC	AllC	rel	dbQ	close
1.00	1.57	1.24	0.33	1.32	0.31	1.22	1.00

This obviously results in a decrease of the number of database queries involving a proportional decrease of the number of system turns N_S . Yet, an increase of the number of user turns N_U is also observed. By examining the action frequencies, one

can notice that the number of constraining questions increased resulting in an increase of N_U . Indeed, the learned strategy implies gathering enough information from the user before performing a database query. This explains why the systems ask more constraining questions.

This last strategy is actually optimal for the considered simulation environment (constant word error rate for all tasks) and is suitable for using with this simple application.

5 Conclusion

In this paper, we first described the paradigms of the Markov Decision Processes (MDP) and of Reinforcement Learning (RL) and explained how they could be used for spoken dialogue strategy optimization. Although RL seems suitable for this task, the parameterization of such a method influences a lot the results and is very task dependent. We therefore wanted to show by three experiments on a very simple database querying system the influence of parameterization. From this, one can say first that the state space representation is a crucial point since it embeds the knowledge of the system about the interaction. Second, the reward (or cost) function is also of major importance since it measures how well the system performs on the task. Performance measure is a key of RL. The three experiments described in the last section showed the influence of these parameters on the learned strategy and concluded that a correctly parameterized RL algorithm could result in an acceptable dialogue strategy while little changes in the parameters could lead to silly strategies unsuitable for use in real conditions.

References

1. Sutton, R. S., Barto, A.G.: Reinforcement Learning : An Introduction. MIT Press (1998)
2. Levin, E., Pieraccini, R., Eckert, W.: Learning Dialogue Strategies within the Markov Decision Process Framework. Proceedings of ASRU'97, Santa Barbara, California (1997)
3. Singh, S., Kearns, M., Litman, D., Walker, M.: Reinforcement Learning for Spoken Dialogue Systems. Proceedings of NIPS'99, Denver, USA (1999)
4. Scheffler, K., Young, S.: Corpus-Based Dialogue Simulation for Automatic Strategy Learning and Evaluation. Proceedings of NAACL Workshop on Adaptation in Dialogue Systems (2001)
5. Pietquin, O., Dutoit, T.: A Probabilistic Framework for Dialog Simulation and Optimal Strategy Learning. IEEE Transactions on Audio, Speech and Language Processing, Volume 14, Issue 2 (2006) 589-599.
6. Walker, M., Litman, D., Kamm, C., Abella, A.: PARADISE: A Framework for Evaluating Spoken Dialogue Agents. Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, Madrid, Spain (1997) 271-280.
7. Pietquin, O., Beaufort, R.: Comparing ASR Modeling Methods for Spoken Dialogue Simulation and Optimal Strategy Learning. Proceedings of Interspeech/Eurospeech 2005, Lisbon, Portugal (2005)

Exploring an Unknown Environment with an Intelligent Virtual Agent

In-Cheol Kim

Department of Computer Science, Kyonggi University
Suwon-si, Kyonggi-do, 442-760, South Korea
kic@kyonggi.ac.kr

Abstract. We consider the problem of exploring an unknown environment with an intelligent virtual agent. Traditionally research efforts to address the exploration and mapping problem have focused on the graph-based space representations and the graph search algorithms. In this paper, we propose DFS-RTA* and DFS-PHA*, two real-time graph search algorithms for exploring and mapping an unknown environment. Both algorithms are based upon the simple depth-first search strategy. However, they adopt different real-time shortest path-finding methods for fast backtracking to the last unexhausted node. Through some experiments with a virtual agent deploying in a 3D interactive computer game environment, we confirm completeness and efficiency of two algorithms.

1 Introduction

Suppose that an agent has to construct a complete map of an unknown environment using a path that is as short as possible. An agent has to explore all nodes and edges of an unknown, strongly connected directed graph. The agent visits an edge when it traverses the edge. A node or edge is explored when it is visited for the first time. The goal is to determine a map of the graph using the minimum number R of edge traversals. At any point in time the robot knows (1) all visited nodes and edges and can recognize them when encountered again; and (2) the number of visited edges leaving any visited node. The agent does not know the head of unvisited edges leaving a visited node or the unvisited edges leading into a visited node. At each point in time, the agent visits a current node and has the choice of leaving the current node by traversing a specific known or an arbitrary unvisited outgoing edge. An edge can only be traversed from tail to head, not vice versa.

If the graph is Eulerian, $2m$ edge traversals suffice, where m is the number of edges. This immediately implies that undirected graphs can be explored with at most $4m$ traversals [2]. For a non-Eulerian graph, let the deficiency d be the minimum number of edges that have to be added to make the graph Eulerian. Recently Kwek [7] proposed an efficient depth-first search strategy for exploring an unknown strongly connected graph G with m edges and n vertices by traversing at most $\min(mn, dn^2+m)$ edges. In this paper, we propose DFS-RTA* and DFS-PHA*, two real-time graph search algorithms for exploring and mapping an unknown environment. Both algorithms are based upon the simple depth-first search strategy. However, they adopt different real-time shortest path-finding methods for fast backtracking to the last unexhausted node. Through some experiments with a virtual agent

Table 1. RTA* Algorithm

Function : RTA*(S_NODE, G_NODE)
C_NODE = S_NODE ; best = 0
do
 for each neighbor Y_NODE of C_NODE
 $f = h(Y_NODE, G_NODE) + k(C_NODE, Y_NODE)$
 if $f < \text{best}$, B_NODE = Y_NODE ; best = f
 advanceTo(B_NODE)
 C_NODE = B_NODE
until C_NODE = G_NODE

deploying in a 3D interactive computer game environment, we confirm the completeness and efficiency of two algorithms.

2 Real-Time Search for Shortest Path Finding

State-space search algorithms for finding the shortest path can be divided into two groups: *off-line* and *real-time*. Off-line algorithms, such as the A* algorithm [8], compute an entire solution path before executing the first step in the path. Real-time algorithms, such as the RTA*(Real-Time A*)[6], perform sufficient computation to determine a plausible next move, execute that move, then perform further computation to determine the following move, and so on, until the goal state is reached. These real-time or on-line algorithms direct an agent to interleave planning and actions in the real world. These algorithms can not guarantee to find the optimal solution, but usually find a suboptimal solution more rapidly than off-line algorithms. The RTA* algorithm shown in Table 1 calculates $f(x')=h(x')+k(x,x')$ for each neighbor x' of the current state x , where $h(x')$ is the current heuristic estimate of the distance from x' to the goal state, and $k(x,x')$ is the distance between x and x' . And then the algorithm moves to a neighbor with the minimum $f(x')$ value. The RTA* revises a table of heuristic estimates of the distances from each state to the goal state during the search process. Therefore, the algorithm is guaranteed to be complete in the sense that it will eventually reach the goal, if certain conditions are satisfied. Fig. 2 illustrates the search process of the RTA* algorithm on the example graph of Fig. 1.

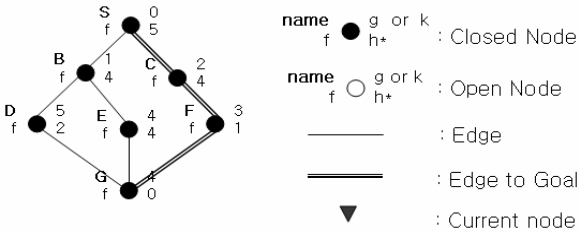


Fig. 1. An Example Graph

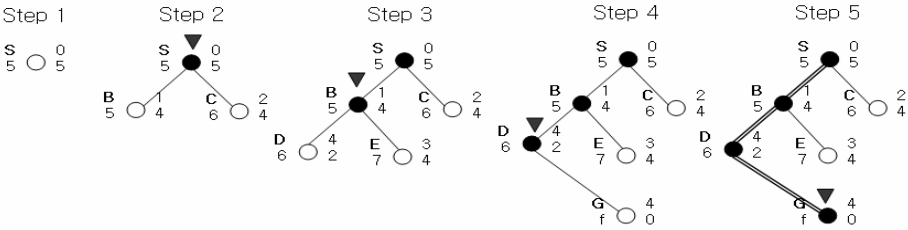


Fig. 2. Search Directed by RTA*

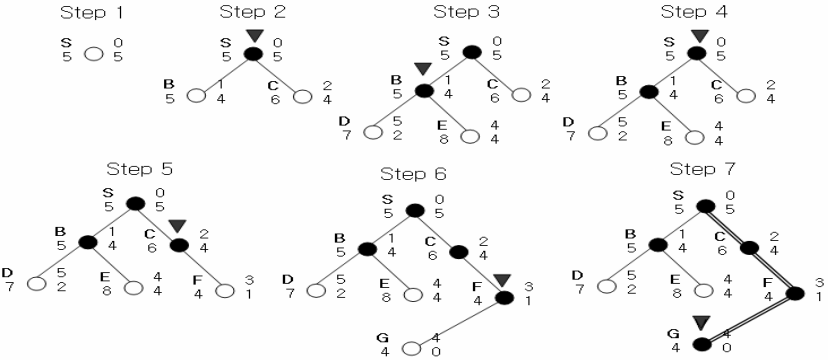


Fig. 3. Search Directed by PHA*

Table 2. PHA* Algorithm

Function : PHA*(S_NODE, G_NODE) C_NODE = S_NODE ; best = 0 do N_NODE = best node from OPEN_LIST if N_NODE = explored lowerLevel(C_NODE , N_NODE) C_NODE = N_NODE until C_NODE = G_NODE Function : lowerLevel(C_NODE, G_NODE) S_NODE = C_NODE f = 0 do for each neighbor Y_NODE of C_NODE f = g(Y_NODE) + h(Y_NODE,G_NODE) if f < best, B_NODE = Y_NODE ; best = f moveTo(B_NODE) C_NODE = B_NODE until C_NODE = G_NODE

The PHA* algorithm [5], which is another real-time search algorithm for finding the shortest path, is a 2-level algorithm. As summarized in Table 2, the upper level is a regular A* algorithm [8], which chooses at each cycle which node from the open list to expand. It usually chooses to expand the node with the smallest f -value in the open list, regardless of whether the agent has visited that node before or not. On the other hand, the lower level directs the agent to that node in order to explore it.

The lower level must use some sort of a navigational algorithm such as Positional DFS (P-DFS), Directional DFS (D-DFS), or A*DFS. Due to the A* algorithm used in the upper level, this PHA* algorithm can guarantee to find the optimal solution. Fig. 3 illustrates the search process of the PHA* algorithm on the example graph of Fig. 1.

3 Search Algorithms for Exploration and Mapping

We consider real-time search algorithms based upon a simple depth-first strategy like Kwek's. Table 3 summarizes the simple depth-first search strategy for exploring an unknown environment. It simply traverses an unvisited edge when there is one until the agent is stuck. As the agent traverses the edges in this greedily manner, we push the current node into a stack. When the agent is stuck, we simply pop the stack until either the stack is empty or the popped node y is not exhausted. In the former case, the agent has already traversed all the edges of G . In the latter case, Kwek claimed that there is a path from u that leads to y in the graph G' obtained by the agent's exploration so far. In Kwek's depth-first search strategy, the agent therefore traverses this path to y and repeats the greedy traversal of the unvisited edges. In our depth-first search strategy for exploring and mapping, however, the agent tries to find an optimal path to y by applying a real-time shortest path-finding procedure such as RTA* or PHA*. In other words, the *moveShortestPath* function in the definition of our simpleDFS algorithm can be instantiated with the RTA* or PHA* function defined above. We call the real-time depth-first exploration algorithm using RTA* (or PHA) as *DFS-RTA** (or *DFS-PHA**).

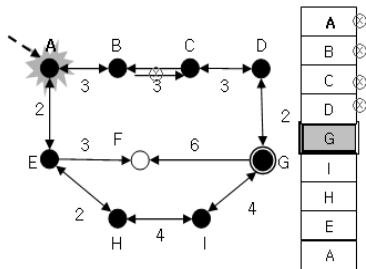


Fig. 4. Search Directed by simpleDFS

Following our search strategy, the agent does not try to find any path to y in the graph G' obtained so far, but find an optimal path by traversing even unexplored nodes and edges outside of G' . Fig. 4 shows an example search directed by our simpleDFS strategy. Suppose the agent has already traversed the cyclic path A-E-H-I-G-D-C-B-A and then it is stuck. It pops the nodes A, B, C, and D from the stack until it

Table 3. Simpledfs Algorithm

```

Function : simpleDFS(S_NODE)
C_NODE = S_NODE  /* start node */
N_NODE, Y_NODE = null
do
  while(exhausted(C_NODE) != true)
    do
      N_NODE = selectNextNode(C_NODE)
      advanceTo(N_NODE)
      push( HISTORY, C_NODE ) /* history stack */
      C_NODE = N_NODE
    end
  do
    Y_NODE = pop(HISTORY)
    until((exhausted(Y_NODE) != true) or
      (empty(HISTORY) = true))
    if (exhausted(Y_NODE) != true),
      moveShortestPath(C_NODE, Y_NODE)
      C_NODE = Y_NODE
    until (empty(HISTORY) = true)

Function : exhausted(C_NODE )
for each neighbor Y_NODE of C_NODE
  if unvisited(Y_NODE), return false
return true

Function : selectNextNode(C_NODE)
f = 10000  /* a large value */
for each neighbor Y_NODE of C_NODE
  if distance(C_NODE, Y_NODE) < f,
    B_NODE = Y_NODE
    f = distance(C_NODE, Y_NODE)
return B_NODE

```

encounters the unexhausted node G. And then it tries to move to the node G through the shortest path and traverses the unexplored edge to F.

4 Implementation

In order to test our exploration algorithms, we implemented an intelligent virtual agent called UTBot. The UTBot is a bot client of the Gamebots system. The Gamebots [1] is a multi-agent system infrastructure derived from Unreal Tournament (UT). Unreal Tournament (UT) is a category of video games known as first-person shooters, where all real time players exist in a 3D virtual world with simulated physics. The Gamebots allows UT characters to be controlled over client-server network connections by feeding sensory information to bot clients and delivering action commands issued from bot clients back to the game server. In a dynamic virtual environment built on the Gamebots system, the UTBot must display human-level capabilities to play successfully, such as learning a map of their 3D environment, planning paths, and coordinating team activities under considering their adversaries. In order to assist

UT characters' move in a 3D virtual world, the UT game maps contains many waypoints perceivable by characters on itself. These waypoints and links connecting two adjacent waypoints constitute a directional graph model for UTBot's exploration. The UTBot was built upon a generic behavior-based agent architecture, CAA (Context-sensitive Agent Architecture). Our CAA consists of (1) a world model; (2) an internal model; (3) a behavior library; (4) a set of sensors and effectors; and (5) an interpreter. Fig. 5 shows the UML class diagram of the UTBot. The UTBot has a variety of external behaviors such as Explore, Attack_Point, Defend_Point, Collect_Powerup, Collect_Weapon, Collect_Armor, Chase, Attack, Retreat, and MoveTo. Table 4 lists available internal modes and the associated external behaviors. To transit from one internal mode to another, the CAA UTBot has a set of internal behaviors such as ExploreToDominate, DominateToCollect, and CollectToDominate.

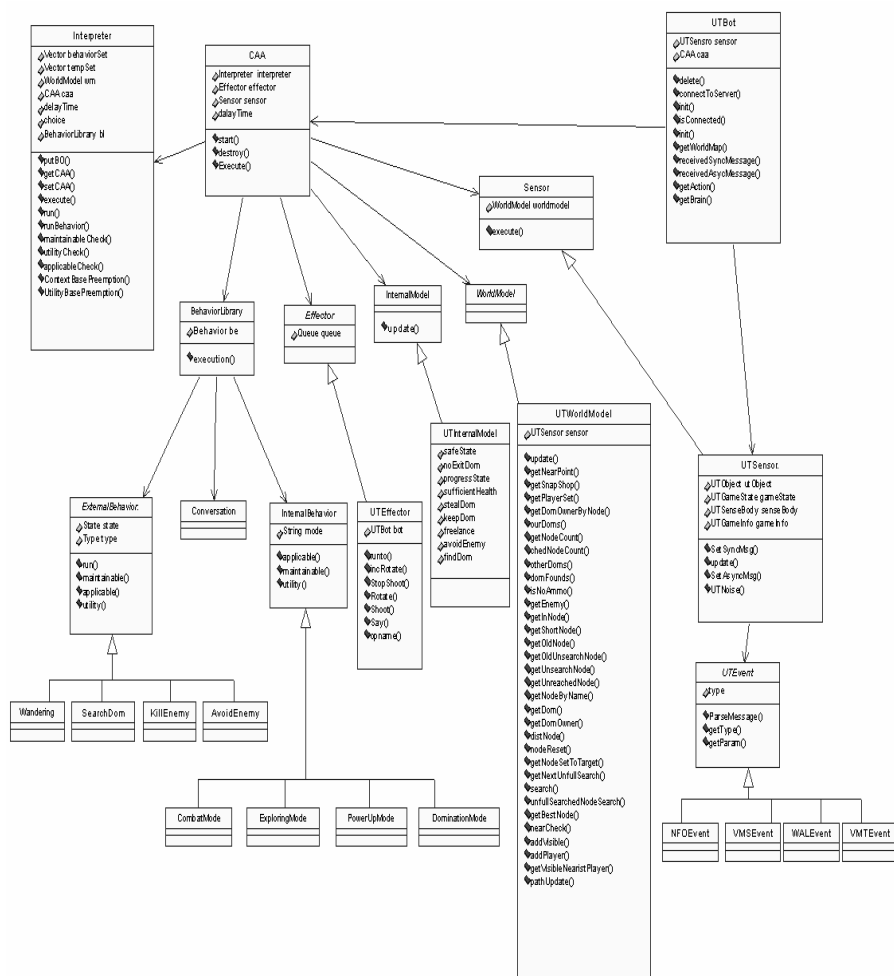


Fig. 5. UML Class Diagram of the UTBot

Table 4. The Internal Modes and the Associated External Behaviors

Internal Modes	External Behaviors
Explore	MoveTo, Explore, Attack, Chase, Retreat
Dominate	MoveTo, Attack_Point, Defend_Point
Collect	MoveTo, Collect_Powerup, Collect_Weapon, Collect_Armor, Retreat, Attack
Died	No Behaviors
Healed	No Behaviors

Based on the proposed DFS-RTA* and DFS-PHA* algorithms, we implemented the UTBot’s exploring behavior. The UTBot provides a visualization tool shown in Fig. 6. This visualization tool can help us keep track and analyze the UTBot’s exploring behavior.

5 Experiments

In order to compare with our DFS-RTA* and DFS-PHA*, we implemented another depth-first search algorithm for exploration. The DFS-DFS algorithm uses a simple depth-first search strategy not only for traversing the entire graph systematically before stuck, but also for finding a backtracking path to the unexhausted node from the obtained graph G’. Four different UT game maps of 90, 120, 150, and 180 nodes were prepared for experiments. On each game map, we tried individual exploration algorithms (DFS-DFS, DFS-RTA*, DFS-PHA*) five times with different starting point. Fig. 7, 8, and 9 represent the result of experiments. Fig. 7 shows the average number of visited nodes, Fig. 8 shows the average traveled distance, and Fig. 9 shows the average consumed time of individual exploration algorithms, respectively.

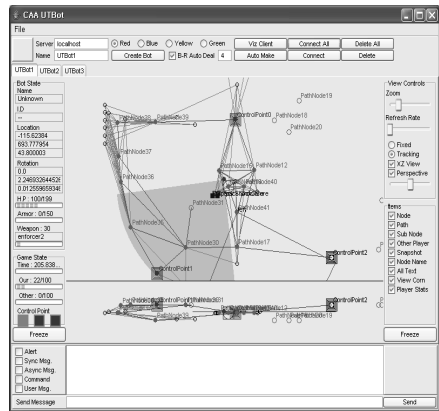


Fig. 6. Visualization Tool

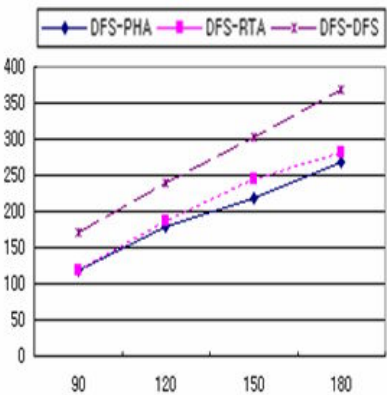


Fig. 7. Visited Nodes

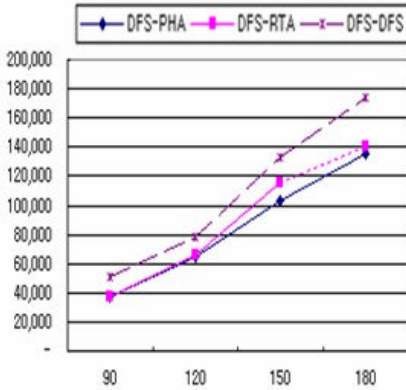


Fig. 8. Traveled Distance

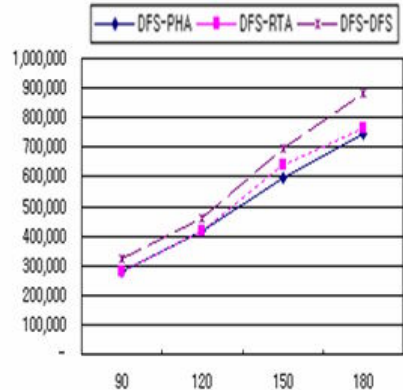


Fig. 9. Consumed Time

We can find out that our DFS-RTA* and DFS-PHA* algorithms outperforms the simple DFS-DFS algorithm in all three different performance measures. In particular, two algorithms show high efficiency in terms of the visited nodes and the traveled distance. Moreover, the more complex and larger world maps are used, the clearer superiority of our algorithms is. We also notice that DFA-PHA* slightly outperforms DFS-RTA* among two proposed algorithms. The reason is in part that the algorithm PHA* can guarantee to find an optimal path to the destination, but the RTA* can guarantee just a suboptimal path.

6 Conclusions

We presented DFS-RTA* and DFS-PHA*, two real-time graph search algorithms for exploring and mapping an unknown environment. Both algorithms are based upon the simple depth-first search strategy, but they use different real-time shortest path-finding methods for fast backtracking to the last unexhausted node. Through some experiments with a virtual agent deploying in a 3D interactive computer game environment, we confirmed the completeness and efficiency of two algorithms.

Acknowledgements. This work was supported by Kyonggi University Research Grant.

References

1. Adobbati, R., Marshall, A.N., Scholer, A., Tejada, S., Kaminka, G.A., Schaffer, S., Sollitto, C.: GameBots: A 3D Virtual World Test Bed for Multiagent Research. Proceedings of the 2nd International Workshop on Infrastructure for Agents, MAS, and Scale MAS (2001)
2. Albers, S. and Henzinger M.: Exploring Unknown Environments. Proceedings the 29th Annual ACM Symposium on Theory Computing, (1997), 416-425.
3. Bender, M., Fernandez, A., Ron, D., Sahai, A., and Vadhan, S.: The Power of a Pebble: Exploring and Mapping Directed Graphs. Proceedings of STOC-98, (1998), 269-278

4. Deng, X., Kameda, T., and Papadimitriou, C.: How to Learn in an Unknown Environment. Proceedings of the 32nd Symposium on the Foundations of Computer Science, (1991), 298-303.
5. Felner A., Stern R., Kraus S., Ben-Yair A., Netanyahu N.S.: PHA*: Finding the Shortest Path with A* in An Unknown Physical Environment. Journal of Artificial Intelligence Research (JAIR), Vol.21 (2004), 631-670
6. Korf, R.E.: Real-time Heuristic Search. Artificial Intelligence, Vol.42, No.3, (1990), 189-211
7. Kwek, S.: On a simple Depth-First Search Strategy for exploring Unknown Graphs. Proceedings of the 5th International Workshop on Algorithms and Data Structures, LNCS 1272, (1997), 345-353
8. Pearl, J. and Kim, J.E.: Studies in Semi-Admissible Heuristics. IEEE Transaction on PAMI, Vol.4, No.201, (1982), 392-400

Case-Based Reasoning Within Semantic Web Technologies

Mathieu d'Aquin^{1,2}, Jean Lieber¹, and Amedeo Napolì¹

¹ LORIA (INRIA Lorraine, CNRS, Nancy University)
Campus scientifique, BP 239, 54 506 Vandœuvre-lès-Nancy, France

² Knowledge Media Institute, The Open University
Walton Hall, Milton Keynes, MK7 6AA, United Kingdom
{daquin, lieber, napolì}@loria.fr,
m.daquin@open.ac.uk

Abstract. The semantic Web relies on the publication of formally represented knowledge (ontologies), retrieved and manipulated by software agents using reasoning mechanisms. OWL (Web Ontology Language), the knowledge representation language of the semantic Web, has been designed on the basis of description logics, for the use of deductive mechanisms such as classification and instantiation. Case-Based reasoning is a reasoning paradigm that relies on the reuse of cases stored in a case base. This reuse is usually performed by the adaptation of the solution of previously solved problems, retrieved from the case base, thanks to analogical reasoning. This article is about the integration of case-based reasoning into the semantic Web technologies, addressing the issue of analogical reasoning on the semantic Web. In particular, we show how OWL is extended for the representation of adaptation knowledge, and how the retrieval and adaptation steps of case-based reasoning are implemented on the basis of OWL reasoning.

Keywords: Case-based reasoning, semantic Web, OWL, description logic reasoning, oncology.

1 Introduction

The case-based reasoning (CBR) process relies on three types of knowledge: the domain knowledge, the adaptation knowledge and the cases [1]. Regarding the semantic Web, the goal of an ontology is to formalize the knowledge about a particular application domain. Thus, an ontology may play the role of the domain knowledge container for CBR. But the semantic Web technologies and the representation languages such as OWL [2] do not include the features required for the representation of knowledge about similarity and adaptation that constitute the core of adaptation knowledge. For this purpose, this article describes an extension of OWL according to a representation model for adaptation knowledge. This model can be seen as an ontology for CBR, allowing to formalize domain-dependent adaptation knowledge to be applied within a domain-independent CBR mechanism. Being represented in OWL, adaptation knowledge can be related with ontologies and cases available on the semantic Web. A CBR service implemented on the basis of this model and of standard OWL inferences is described hereafter, and can be considered as a new reasoning tool for adaptation on the semantic Web.

Our main motivation for the integration of CBR within the semantic Web infrastructure is the development a semantic portal dedicated to knowledge management and decision support in oncology. This application is briefly described in the next section. Section 3 presents OWL, the standard language for ontology representation, and so, one of the most important semantic Web technologies. OWL can be used to formalize the domain knowledge and the cases for CBR, but has no facilities for handling adaptation knowledge. The way OWL is extended for adaptation knowledge representation is detailed in section 4. Using OWL as a knowledge and case representation language for CBR allows to apply standard OWL reasoning mechanisms, like subsumption and instantiation, within the CBR inferences, as shown in section 5. Finally, section 6 draws some conclusions and points out the future work.

2 Motivating Application: A Semantic Portal in Oncology

In the Lorraine region of France, the adequate therapeutic decision is established for the majority of the patients by applying a medical protocol that associates standard patient characteristics with a recommended treatment. Even if it is designed to take into account the majority of the medical cases, a protocol does not cover all the situations. Decisions concerning out of the protocol patients are elaborated within a multi-disciplinary expert committee, and rely on the adaptation of the solutions provided by the protocol for similar cases. OWL has been used for the formalization of the knowledge contained in a protocol. Furthermore, in order to provide an intelligent access to knowledge for distant practitioners, a semantic portal has been built, relying on the reasoning mechanisms associated with OWL, and providing a support for knowledge management and decision making in oncology [3]. In the perspective of decision support for out of the protocol cases, a CBR mechanism may be applied on formalized protocols [4]. For this reason, the knowledge used by expert committees may be represented and operationalized as adaptation knowledge, in a declarative way, in order to become sharable and reusable.

3 A Brief Introduction to OWL

This section presents an overview of the OWL language with the aim of introducing the basic definitions that are used in the rest of the paper. The complete syntax and semantics of OWL can be found in [2].

An OWL ontology contains definitions of classes, properties and individuals from the represented domain. An *individual* corresponds to an object. A *property* denotes a binary relation between objects. A *class* represents a set of objects. The semantics of an OWL ontology is given by an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non empty set, called the *interpretation domain*, and $\cdot^{\mathcal{I}}$ is the interpretation function. This function maps a class C into a subset $C^{\mathcal{I}}$ of the interpretation domain $\Delta^{\mathcal{I}}$, a property p into a subset $p^{\mathcal{I}}$ of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and an individual a to an element $a^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$.

An OWL *ontology* O is defined by a set of axioms and a set of assertions. Classes are introduced through the use of *axioms* of the form¹ $C \sqsubseteq D$, C and D being two classes.

¹ In this paper, we use the description logic way of writing expressions in OWL.

$C \sqsubseteq D$ is satisfied by an interpretation \mathcal{I} if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. $C \equiv D$ is a notation for $C \sqsubseteq D$ and $D \sqsubseteq C$. Axioms of the form $p \sqsubseteq q$ between properties (p and q) also exist in OWL. *Assertions* are used to introduce individuals. There are two types of assertions: $C(a)$ (a is an instance of C) and $p(a, b)$ (a and b are related by p), C being a class, a and b two individuals, and p a property. $C(a)$ is satisfied by an interpretation \mathcal{I} if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and $p(a, b)$ is satisfied by \mathcal{I} if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in p^{\mathcal{I}}$. \mathcal{I} is a model of an ontology O whenever it satisfies all the axioms and assertions defining O . OWL provides constructors for building complex classes like the class conjunction $((C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}})$ and the existential quantifier, $((\exists p.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y \in C^{\mathcal{I}}, (x, y) \subseteq R^{\mathcal{I}}\})$.

4 Adaptation Knowledge Representation Within OWL Ontologies

CBR is mainly based on two operations: retrieval and adaptation. The *retrieval* operation selects a source problem *srce* that is considered to be similar to the target problem *tgt* to be solved. The problem *srce* retrieved in the case base is associated with a solution $Sol(srce)$. The goal of *adaptation* is to modify $Sol(srce)$ in order to build a solution $Sol(tgt)$ to *tgt*.

In knowledge-intensive CBR [5], case representation and CBR inferences rely on *domain knowledge* and on *adaptation knowledge*, i.e. knowledge about similarity and adaptation. On the semantic Web, domain knowledge is contained in OWL ontologies. A CBR mechanism on the semantic Web has to be able to manage reusable and sharable adaptation knowledge, in a technology compatible with the semantic Web infrastructure. The model of reformulations introduced in [6] is a general framework for modeling adaptation knowledge into simple and separated components. In the following, we propose a formalization of the reformulation model in OWL, providing a way to represent and to operationalize adaptation knowledge in relation with semantic Web ontologies.

4.1 Reformulations

A reformulation is a pair (r, \mathcal{A}_r) where r is a relation between problems, and \mathcal{A}_r is an *adaptation function*: if r relates *srce* to *tgt*—denoted by “*srce* r *tgt*”—then any solution $Sol(srce)$ of *srce* can be adapted into a solution $Sol(tgt)$ of *tgt* thanks to the adaptation function \mathcal{A}_r —denoted by “ $Sol(srce) \mathcal{A}_r Sol(tgt)$ ”.

In the reformulation model, retrieval consists in finding a *similarity path* relating *srce* to *tgt*, i.e. a composition of relations r_k , introducing intermediate problems pb_k between the source and the target problems. Every r_k relation is linked by a reformulation to an adaptation function \mathcal{A}_{r_k} . Accordingly, the sequence of adaptation functions following the similarity path is reified in a corresponding *adaptation path* (see figure 1).

The model of reformulations is a general framework for representing adaptation knowledge. The operations corresponding to problem relations r_k and adaptation functions \mathcal{A}_{r_k} have to be designed for a particular application or purpose. Most of the time, these operations rely on transformation operations such as specialization, generalization and substitution. These transformations allow the creation of the pb_k problems for

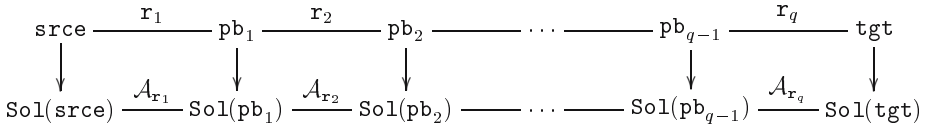


Fig. 1. A similarity path from $srce$ to tgt (first line) and the corresponding adaptation path (second line)

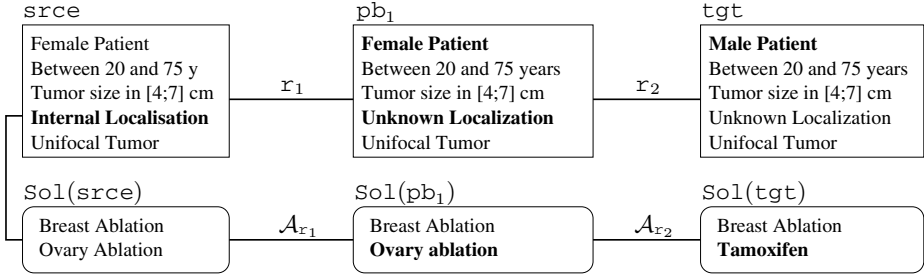


Fig. 2. An example of problem-solving using reformulations

building the similarity path and of the $Sol(pb_k)$ solutions for the adaptation path: a relation of the form $pb_1 \ r \ pb_2$ and an adaptation such as $Sol(pb_1) \ \mathcal{A}_r \ Sol(pb_2)$ both correspond to an application of a transformation.

Moreover, the reformulation framework follows the principle of adaptation-guided retrieval [7], i.e. only source cases for which a solution is adaptable are retrieved, meaning (and implying) that adaptation knowledge is available. According to this principle, similarity paths provide a “symbolic reification” of similarity between problems, allowing the case-based reasoner to build understandable explanation of the results.

An Example Application of Reformulations to Breast Cancer Treatment. In our application to oncology, CBR is used for building recommendations for patients for which the protocol does not provide any satisfactory answer: the recommendations provided by the protocol for similar patients are adapted to these out of the protocol patients. In this application, the problems are descriptions of patients and the solutions are descriptions of the treatment to apply. Figure 2 presents a simplified and informal example of the application of reformulations in this domain. In this example, the tgt problem corresponds to the description of a patient for which the protocol cannot be applied directly. There are two reasons for that. First, this patient is a man, and the protocol for breast cancer treatment is designed to take into account only female patients. Second, for this patient, the tumor localization cannot be determined in the breast (linked with the fact that the patient is a man). The other characteristics used for taking a decision concern the age of the patient, the size of the tumor and the fact that the tumor has only one focus point. Two reformulations, (r_1, \mathcal{A}_{r_1}) and (r_2, \mathcal{A}_{r_2}) , are applied for solving the tgt problem. The first one indicates that, when the localization of the tumor is unknown, the patient should be considered as if he had an internal tumor (because it is the worst case, and so, the one for which the largest treatment should be applied).

The adaptation function \mathcal{A}_{r_1} simply corresponds to a copy of the solution. The second reformulation expresses that, when the difference between the target problem tgt and another problem pb_1 relies on different values for the sex (male for pb_1 and female for tgt), then the adaptation of $\text{Sol}(\text{pb}_1)$ for building $\text{Sol}(\text{tgt})$ consists in replacing the ovary ablation (not applicable to a man) by another hormone therapy having similar effects, i.e. some cures of an anti-oestrogen drug called tamoxifen.

In the following, the above example will be used to illustrate our approach and will be formalized in OWL. It has obviously been largely simplified and many points could be discussed concerning the quality of the knowledge modeling. However, these considerations have no influence on the validity of the presented general approach.

4.2 Reformulations in OWL

CBR relies on the notions of problem, solution, similarity and adaptation. In the reformulation model, similarity and adaptation are reified through the relations r between problems, the adaptation functions \mathcal{A}_r , the similarity paths and the adaptation paths. In this section, we show how to integrate these elements within the semantic Web infrastructure and OWL ontologies. In other terms, we build a *CBR ontology* in OWL, on the basis of the reformulation model. It can be noticed that, although the reformulation model is simple, its representation is based on advanced knowledge representation features such as, for example, property reification. The OWL representation of the reformulation model is depicted in figure 3, and detailed hereafter.

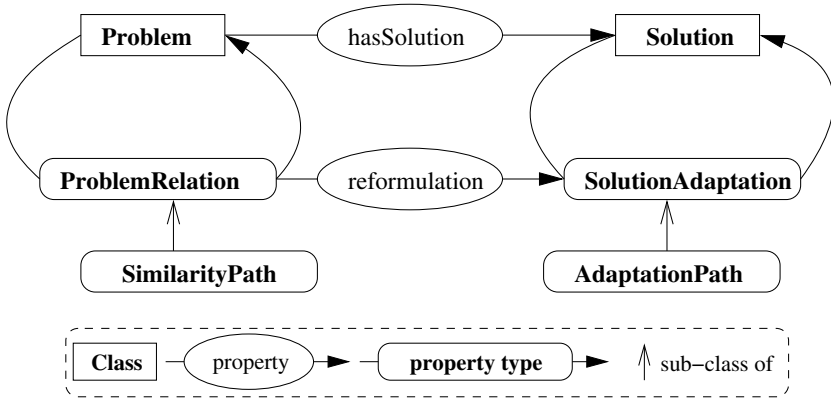


Fig. 3. Schema of the reformulation model in OWL

Problems and Solutions. CBR operations manipulate cases of the form $(\text{pb}, \text{Sol}(\text{pb}))$, corresponding to problems pb associated with their solutions $\text{Sol}(\text{pb})$. Using the OWL model of figure 3, a problem is represented by an instance of the `Problem` class and a solution by an instance of the `Solution` class. Particular problems are associated with their solutions using a property called `hasSolution`. In this way, a CBR service relying on the reformulation ontology can be applied on case bases represented in OWL by a set of assertions of the form:

Problem(p) Solution(s) hasSolution(p,s)

The target problem can be simply declared as an instance of the Problem class:

Problem(tgt)

Additionally, each problem and solution are described by assertions on elements of the domain ontology. The goal of the CBR process is then to relate the tgt individual to instances of the Solution class by the hasSolution property.

The classes Problem and Solution play a central role in the relationship between the CBR process, involving elements of the reformulation model, and the domain knowledge, represented in a domain ontology. In the application for breast cancer treatment, the domain knowledge and the case base are contained in an OWL representation of the decision protocol. This protocol associates instances of the Patient class with instances of the Treatment class using the recommendation property. Thus, in order to apply the CBR service on this representation the following axioms are declared:

Patient \sqsubseteq Problem Treatment \sqsubseteq Solution
 recommendation \sqsubseteq hasSolution

Similarity and Adaptation. In the reformulation model, a relation r represents a link between two problems and can be considered as a knowledge element for modeling the similarity in a particular application domain. A relation r is formalized in OWL by a property holding between two problems. For example, the relation corresponding to a change of sex in figure 2 (the r_2 relation) corresponds to a property linking an instance of female patient to an instance of male patient. In OWL, one can indicate on which class a property should apply, i.e. the *domain* of the property. $\text{domain}(p, C)$ is a notation indicating that the class C represents the domain of the property p . In the same way, the range of a property is the class in which the property should take its values. The notation for indicating that a class C represents the range of a property p is $\text{range}(p, C)$. Thus, the two relations r_1 and r_2 used in the example figure 2 can be introduced in the following way:

$\text{domain}(r1, P\text{-LInt})$ $\text{range}(r1, P\text{-LUnk})$
 $\text{domain}(r2, \text{Patient-F})$ $\text{range}(r2, \text{Patient-M})$

where the four involved classes are defined in the domain knowledge (the protocol representation) by the following axioms:

$P\text{-LUnk} \equiv \text{Patient} \sqcap \exists \text{hasTumor} . (\exists \text{localization} . \text{Unknown})$
 $P\text{-LInt} \equiv \text{Patient} \sqcap \exists \text{hasTumor} . (\exists \text{localization} . \text{Internal})$
 $\text{Patient-M} \equiv \text{Patient} \sqcap \exists \text{sex} . \text{Male}$
 $\text{Patient-F} \equiv \text{Patient} \sqcap \exists \text{sex} . \text{Female}$

In the same way, the adaptation functions \mathcal{A}_r are represented by properties, having subclasses of Treatment as domain and range.

Relations r between problems and adaptation functions \mathcal{A}_r are introduced using two new *types of properties*, i.e. `ProblemRelation` and `SolutionAdaptation`. Types of properties are subclasses of the `Property` class in OWL². The relations between problems and the adaptation functions of the previous example (r_1 , r_2 , \mathcal{A}_{r_1} and \mathcal{A}_{r_2}) are represented by properties, instances of the `ProblemRelation` and `SolutionAdaptation` property types, and introduced by the following assertions:

```

ProblemRelation(r1)          ProblemRelation(r2)
SolutionAdaptation(Ar1)      SolutionAdaptation(Ar2)

```

In the reformulation model, a similarity path is defined as a sequence of relations r_i between problems, relating a source problem to a target problem, and introducing intermediary problems pb_i . Therefore, a similarity path can also be considered as a relation between problems. The property type `SimilarityPath` is then declared as a subclass of `ProblemRelation`. The chaining of the relations r_i contained in a similarity path is represented on the basis of a recursive definition using three properties: `previousRelation`, `pbi` and `nextRelation`. For example, the similarity path

srce r_1 pb₁ r_2 pb₂ r_3 tgt

can be described in OWL by the following assertions:

```

r1(srce, pb1)  SimilarityPath(cs1)      SimilarityPath(cs2)
r2(pb1, pb2)   previousRelation(cs1, r1) previousRelation(cs2, r2)
r3(pb2, tgt)   pbi(cs1, pb1)            pbi(cs2, pb2)
cs1(srce, tgt) nextRelation(cs1, cs2)     nextRelation(cs2, r3)
cs2(pb1, tgt)

```

In the same way, adaptation paths are represented using the `AdaptationPath` property type, which is a subclass of `SolutionAdaptation`. They correspond to recursive definitions of sequences of adaptation functions \mathcal{A}_{r_i} .

Reformulations. A reformulation is a pair (r, \mathcal{A}_r) , associating a relation r between problems to an adaptation function \mathcal{A}_r . Therefore, a reformulation is represented in the OWL model by a property occurring between instances of `ProblemRelation` and instances of `SolutionAdaptation`. The two reformulations (r_1, \mathcal{A}_{r_1}) and (r_2, \mathcal{A}_{r_2}) of the above example are expressed by the following assertions:

```

reformulation(r1, Ar1)      reformulation(r2, Ar2)

```

Moreover, in order to respect the principle of *adaptation guided retrieval* (a problem is retrieved only if its solution is adaptable), a constraint is declared on the `ProblemRelation` class, through the axiom

² Here, we are using some advanced features of OWL: meta-classes and reflexivity. OWL has three nested sub-languages –OWL Lite, OWL DL and OWL Full– and these features are available only in the most expressive one: OWL Full.

$\text{ProblemRelation} \sqsubseteq \exists \text{reformulation.SolutionAdaptation}$

indicating that any `ProblemRelation` should necessary be associated with at least one `SolutionAdaptation` in a reformulation. The adaptation operation of the CBR process is in charge of associating the corresponding adaptation path to any similarity path built during the retrieval operation.

5 OWL Reasoning Within CBR Inferences

OWL has been designed on the principles of description logics. The inference mechanisms usually available with these formalisms (namely subsumption and instantiation) are useful when considering the implementation of a CBR system [8].

Given two classes C and D , the *subsumption test* in OWL is defined by C is subsumed by D (C is more specific than D) if, for every model \mathcal{I} of O , $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. Based on the subsumption test, *classification* consists in finding for a class C , the classes subsuming C and the classes subsumed by C . Classification organizes the classes of the ontology in a hierarchy. Regarding CBR, the class hierarchy is used as a indexing structure for the case base, where a class represents an index for a source problem.

An index is an abstraction of a source problem, containing the relevant part of the information leading to a particular solution [9]. We define an index $\text{idx}(\text{srce})$ as a *class for which srce is an instance* and such that *the solution Sol(srce) of srce can be applied to solve any problem recognized as being an instance of idx(srce)*. In other terms, whenever a problem tgt is recognized as an instance of $\text{idx}(\text{srce})$, index of srce , the solution $\text{Sol}(\text{srce})$ of srce can be reused (without modification) to solve tgt , i.e. $\text{Sol}(\text{srce}) = \text{Sol}(\text{tgt})$.

In OWL, *instance checking* tests whether an individual a is an instance of a class C , i.e. if for every model \mathcal{I} of O , $a^{\mathcal{I}} \in C^{\mathcal{I}}$. It supports the *instantiation* reasoning service that consists, given an individual a , in finding the classes of which a is an instance. Using instantiation, the retrieval operation of CBR consists in finding the index of a source problem having an adaptable solution, rather than the source problem directly. Thus, the similarity path built during retrieval relates the tgt problem to a problem pb_0 , such that pb_0 is an instance of $\text{idx}(\text{srce})$:

$$\text{srce} \xrightarrow{\text{isa}} \text{idx}(\text{srce}) \xleftarrow{\text{isa}} \text{pb}_0 \text{ } r_1 \text{ } \text{pb}_1 \text{ } r_2 \dots \text{pb}_{q-1} \text{ } r_q \text{ } \text{tgt}$$

where an “isa” arrow means “is an instance of”. Since pb_0 is recognized as an instance of $\text{idx}(\text{srce})$, its solution $\text{Sol}(\text{pb}_0)$ corresponds to the solution $\text{Sol}(\text{srce})$ of srce . Therefore, the corresponding adaptation path is:

$$\text{Sol}(\text{srce}) = \text{Sol}(\text{pb}_0) \text{ } \mathcal{A}_{r_1} \text{ } \text{Sol}(\text{pb}_1) \text{ } \mathcal{A}_{r_2} \dots \text{Sol}(\text{pb}_{q-1}) \text{ } \mathcal{A}_{r_q} \text{ } \text{Sol}(\text{tgt})$$

When using the breast cancer treatment protocol as a case base, source problems are represented by their index classes, i.e. by sub-classes of the `Patient` class. Solutions are then directly attached the these index classes, through axioms of the form: $P \sqsubseteq \exists \text{recommendation.T}$, where P is a subclass of `Patient` and T is a subclass of

treatment. Concerning the example in figure 2, the source problem is then represented by a index class I , linked to the recommended treatment class S in the following way:

$$\begin{aligned} I &\equiv \text{Patient-F} \sqcap \exists \text{age.ge20} \sqcap \exists \text{age.le75} \\ &\quad \sqcap \exists \text{hasTumor}(\text{NonMultifocalTumor} \sqcap \exists \text{localization.Internal} \sqcap \\ &\quad \exists \text{size.ge4} \sqcap \exists \text{size.le7}) \\ S &\equiv \text{BreastAblation} \sqcap \text{OvaryAblation} \\ I &\sqsubseteq \exists \text{recommendation.S} \end{aligned}$$

where ge20 represents the integer values greater or equal than 20, le75 the integer values lower or equal than 75, ge4 the float values greater or equal than 4 and le7 the float values lower or equal than 7.

On the basis of the previously defined relations r_1 and r_2 between problems, the following similarity path is then built by the retrieval operation using OWL instantiation:

$$I \xleftarrow{\text{isa}} \text{pb}_0 \quad r_1 \quad \text{pb}_1 \quad r_2 \quad \text{tgt}$$

tgt being an instance of Patient having the properties corresponding to the characteristics of the patient to be treated (male sex, unknown localization, etc.). pb_0 and pb_1 are also instances of Patient . According to r_2 , pb_1 has the same properties as tgt except the sex (male for tgt and female for pb_1). According to r_1 , pb_0 has the same properties as pb_1 except the localization of the tumor (unknown for pb_1 , internal for pb_0). pb_0 is recognized as an instance of the I index class defined previously, and is thus associated with a solution $\text{Sol}(\text{pb}_0)$ of the S treatment class (i.e. a breast ablation and an ovary ablation). Finally, the adaptation of this solution is based on the following adaptation path:

$$S \xleftarrow{\text{isa}} \text{Sol}(\text{pb}_0) \quad \mathcal{A}_{r_1} \quad \text{Sol}(\text{pb}_1) \quad \mathcal{A}_{r_2} \quad \text{Sol}(\text{tgt})$$

where $\text{Sol}(\text{pb}_1)$ is a copy of $\text{Sol}(\text{pb}_0)$ (application of \mathcal{A}_{r_1}) and $\text{Sol}(\text{tgt})$ is built by replacing the ovary ablation in $\text{Sol}(\text{pb}_1)$ by some cures of tamoxifen (application of \mathcal{A}_{r_2}): $\text{Sol}(\text{tgt}) \xrightarrow{\text{isa}} \text{BreastAblation} \sqcap \text{Tamoxifen}$.

Instantiation is also used to infer new knowledge units about an individual on the basis of its concept membership, and of constraints contained in concept definitions. For example, when a patient pat is an instance of the concept MalePatient , and if it is declared that the localization of the tumor cannot be established for a man, i.e.

$$\text{MalePatient} \sqsubseteq \forall \text{tumor}(\exists \text{localisation.Undeterminable})$$

then it can be inferred that the localization of the tumor is Undeterminable for pat . This mechanism is used to help the user elaborating the target problem for CBR.

6 Conclusion

This article describes a semantic Web based model for adaptation knowledge representation, and its application in a CBR service, reusing the knowledge published on

the semantic Web (OWL ontologies) and the inferences associated with OWL (subsumption and instantiation). Some studies have been interested in defining markup languages for case representation, on the basis of XML [10] or RDF [11]. Description logics have also been used for knowledge intensive CBR in several systems (see e.g. [8]). The research work presented here consider the building of CBR systems in the semantic Web framework and can be seen as a first guideline for practitioners to apply such techniques. Moreover, a prototype CBR system relying on OWL has been implemented and is currently tested on our application in oncology. Although it has been developed for the purpose of medical protocol adaptation, this system is generic in the sense that it is independent of the application domain and can be reused in any domain where domain knowledge, adaptation knowledge and cases are formalized in OWL.

This last point, having OWL knowledge available, leads to a particularly important ongoing work: *adaptation knowledge acquisition*. In the CBR literature, only a few papers investigate this issue (see e.g. [12]). For the purpose of the application in oncology, the acquisition of adaptation knowledge on the basis of dialogs with experts has already been experimented [13]. A tool applying the principles of knowledge discovery in databases and data mining is also currently developed [14].

References

1. Lenz, M., Bartsch-Spörl, B., Burkhard, H.D., Wess, S., eds.: Case-Based Reasoning Technology: From Foundations to Applications, LNAI 1400. Springer (1998)
2. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D., Patel-Schneider, P., Stein, L.A.: OWL Web Ontology Language Reference. W3C Recommendation (2004)
3. d'Aquin, M., Brachais, S., Bouthier, C., Lieber, J., Napoli, A.: Knowledge Editing and Maintenance Tools for a Semantic Portal in Oncology. *International Journal of Human-Computer Studies (IJHCS)* **62** (2005)
4. Lieber, J., Bresson, B.: Case-Based Reasoning for Breast Cancer Treatment Decision Helping. (In: Proc. of the Fifth European Workshop on Case-Based Reasoning - EWCBR'2000)
5. Aamodt, A.: Knowledge-Intensive Case-Based Reasoning and Sustained Learning. In Aiello, L.C., ed.: Proc. of the 9th European Conference on Artificial Intelligence. (1990)
6. Melis, E., Lieber, J., Napoli, A.: Reformulation in Case-Based Reasoning. In Smyth, B., Cunningham, P., eds.: Proc. of the European Workshop on Case-Based Reasoning, EWCBR'98, Springer (1998) 172–183
7. Smyth, B.: Case-Based Design. PhD. thesis, Trinity College, University of Dublin (1996)
8. Gómez-Albarrán, M., Gonzàles-Calero, P., Díaz-Agudo, B., Fernández-Conde, C.: Modelling the CBR Life Cycle Using Description Logics. In Althoff, K.D., Bergamnn, R., Branting, L., eds.: Proc. of the International Conference on Case-Based Reasoning, ICCBR'99. Volume 1650 of Lecture Notes in Artificial Intelligence., Springer (1999) 147–161
9. Lieber, J., Napoli, A.: Using classification in case-based planning. In: Proc. of the 12th European Conference on Artificial Intelligence. (1996) 132–136
10. Coyle, L., Doyle, D., Cunningham, P.: Representing Similarity for CBR in XML. In: Procs. of the Seventh European Conference on , LNAI 3155. (2004) 119–127
11. Chen, H., Wu, Z.: CaseML: a RDF-based Case Markup Language for Case-based Reasoning in Semantic Web. In: From structured cases to unstructured problem solving episodes for experience-based assistance. Workshop at ICCBR-2003. (2003)

12. Hanney, K., Keane, M.: Learning Adaptation Rules from Cases. In Smith, I., Falting, B., eds.: Proc. of the 3rd European Workshop on Case-Based Reasoning, EWCBR-96. Volume 1168 of LNAI., Springer (1996)
13. Lieber, J., d'Aquin, M., Bey, P., Napoli, A., Rios, M., Sauvagnac, C.: Acquisition of Adaptation Knowledge for Breast Cancer Treatment Decision Support. In: Proc. of the 9th Conference on Artificial Intelligence in Medicine in Europe, AIME 2003. (2003)
14. d'Aquin, M., Badra, F., Lafrogne, S., Lieber, J., Napoli, A., Szathmary, L.: Adaptation Knowledge Discovery from a Case Base. In: Proc. of the poster session of the European Conference on Artificial Intelligence (to appear). (2006)

A Proposal for Annotation, Semantic Similarity and Classification of Textual Documents

Emmanuel Nauer and Amedeo Napoli

LORIA — UMR 7503
Bâtiment B, B.P. 239
F-54506 Vandœuvre-lès-Nancy cedex, France
{nauer, napoli}@loria.fr

Abstract. In this paper, we present an approach for classifying documents based on the notion of a semantic similarity and the effective representation of the content of the documents. The content of a document is annotated and the resulting annotation is represented by a labeled tree whose nodes and edges are represented by concepts lying within a domain ontology. A reasoning process may be carried out on annotation trees, allowing the comparison of documents between each others, for classification or information retrieval purposes. An algorithm for classifying documents with respect to semantic similarity and a discussion conclude the paper.

Keywords: content-based classification of documents, domain ontology, document annotation, semantic similarity.

1 Introduction and Motivation

In this paper, we propose an approach for defining a semantic annotation of Web textual documents based on the content of the documents. The core of the approach relies on the notions of annotation tree and semantic similarity, allowing to manipulate documents with respect to their content, for, e.g. reasoning and information retrieval. An annotation is represented as a labeled tree according to a domain ontology, and is named annotation tree. Annotation trees can be compared and classified, and are the basis for evaluating a semantic similarity between documents.

Most of the information retrieval systems are based on keyword search, with a more or less sophisticated use of keywords, including for example normalized or weighted keywords, weighted or thesaurus-based relations between keywords for document matching and retrieval [16]. These information retrieval approaches are based on a rather rough and direct use of the set of –unrelated– keywords associated to a document, whereas it could be useful to take into account the semantic relations existing between keywords. These approaches are efficient for simple and standard tasks, but show their limits within more complex applications, e.g. applications for semantic Web, where an actual, explicit, and semantic access to the content of documents is needed. In addition, research

work on information retrieval and semantic Web is mainly based on the use of domain knowledge and reasoning, for improving document representation and query answering. Research work on semantic annotation of documents aims at making inferences using a knowledge-based annotation of documents, turning a human-understandable content into a machine understandable content [10,9,18].

In this research work, we present a content-based classification of textual Web documents based on a semantic annotation of the content of documents. A semantic annotation is represented as a labeled tree, where a node is typed by a concept of the domain ontology, while an edge is typed by a relation between two concepts. The ontology holds on the domain of the documents, and includes concepts and relations organized by a subsumption relation. This kind of labeled-tree representation can be likened to graphs of RDF statements or to the XML object model of documents (DOM), where the semantics associated to the elements of an annotation relies on a domain ontology.

This content-based annotation of documents supports a classification process using semantic similarity between document annotations. The classification process allows to organize documents in categories with respect to their contents and domain knowledge. Semantically similar documents are classified within a class describing their common characteristics according to the domain ontology, where an individual document is reified as an instance of a class. In this way, documents holding on some topics of interest may be classified according to the classes representing these topics in the domain ontology. For example, it can be useful for a researcher looking for specific scientific publications to navigate within a hierarchy of classes, where a class represents a set of documents holding on some selected topics, instead of consulting a flat and unordered list of documents. In this context, a query is represented as a class whose instances (representing individual documents) are considered as answers to the query. For answering a query, the class representing the query is built and then classified in the hierarchy of document categories. Once the query is classified, the instances of the classes subsumed by the query are returned as potential answers to the query.

The paper is organized as follows. In section 2, the notion of semantic annotation of documents is introduced. Then, in section 3, semantic similarity and document classification are detailed and illustrated. A discussion on the annotation and classification processes ends the paper.

2 The Annotation of Textual Web Documents

2.1 Introducing the Domain Ontology

Given a reference domain \mathcal{D} , an ontology $\mathcal{O}_{\mathcal{D}}$ is considered as a hierarchy of classes and relation classes: classes represent the concepts of the domain \mathcal{D} while relation classes represent relations between concepts [13,17]. The ontology $\mathcal{O}_{\mathcal{D}}$ is used for studying a set of documents DOCS related to the domain \mathcal{D} . A class has a name and is either primitive or defined by a set of attributes. As in description logics, the attributes of a defined class act as necessary and sufficient conditions for declaring an individual as an instance of a defined class [3]. In the same way,

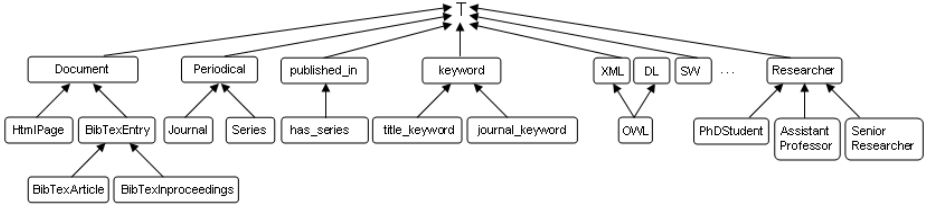


Fig. 1. A fragment of an ontology of classes and relation classes holding on computer science documents

a relation class has a name, a domain, a range, and additional properties such as reflexivity, symmetry, transitivity. . . Classes are organized within a hierarchy, namely \mathcal{O}_D , by a subsumption relation: whenever the class C_1 is subsumed by the class C_2 , written $C_1 \sqsubseteq C_2$, then the individuals that are instances of C_1 are also instances of C_2 . For relation classes, the relation R_1 is subsumed by the relation R_2 , written $R_1 \sqsubseteq R_2$, when, given two individuals a and b , $R_1(a, b)$ implies $R_2(a, b)$. All classes and relation classes are subsumed by \top (Top), the root of the hierarchy \mathcal{O}_D . A fragment of an ontology is given in figure 1, where the relation class **keyword** subsumes the relation class **title_keyword**, and the class **Document** subsumes the classes **HtmlPage** and **BibTexEntry**.

2.2 The Annotation of Documents

Definition 1. An annotation associated to a document D , denoted by $A(D)$, is defined as a labeled rooted tree $A(D) = (N, E)$ where N is a set of nodes having a label and a type, and E is a set of edges having the form $e = (n, a, n')$ where $n, n' \in N$ and a is the label of the edge. The labeled tree $A(D) = (N, E)$ associated D is called the annotation tree of the document D .

In such a tree-based annotation, a node and an edge have a label and a type. The type makes reference either to a class of \mathcal{O}_D or to the special datatype **String** (not explicitly represented in \mathcal{O}_D , and unique datatype considered in the present framework). For notational convenience, the type of an element x is supposed to be returned by the function **type**(x). In addition, the label of a node or an edge is derived from the associated type. Then, for an edge $e = (n, a, n')$, we will indifferently write **type**(e) or **type**(a) for denoting the type of the edge e . A URI, i.e. a *Uniform Resource Identifier*, may be associated to a node for pointing to a specific resource (as in RDF statements). In case the node is a leaf in the annotation tree (i.e. the node has no descendant), a “value” whose type is **String** may be attached to that node. By analogy with types, the value or the URI attached to a node are supposed to be returned by the function **value**(x). It is assumed that a leaf can have a value or a URI, but not both (exclusive or).

At the moment, an annotation is represented as a tree and not as a graph (as it could be the case with RDF statements). One main reason is for keeping things more simple, i.e. a simple structure of an annotation allowing efficient comparison procedures and a simple conceptual representation (using description logics).

The figure 2 gives an example of two documents D_1 and D_2 with their associated annotation trees. The annotation tree associated to D_1 describes an *HTML webpage* about a publication in the *ERCIM News journal*, where the authors, the title, the keywords, the journal keywords... are extracted from the HTML page. In the same way, the tree associated to D_2 describes a *BibTex entry* using the main characteristics of this entry.

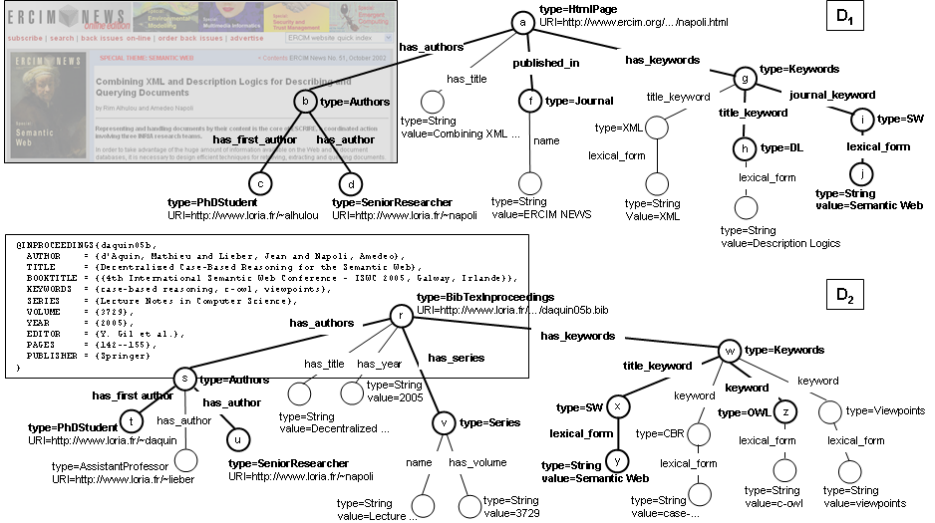


Fig. 2. Example of two documents with their annotation trees

It can be useful to distinguish between *generic* and *specific* annotations. A generic annotation is associated to a set of different documents while a specific annotation is associated to only one particular document. Firstly, a node n is *specific* or *instantiated* whenever it has an associated URI or a value (whose type is **String**); otherwise the node is *generic*. Accordingly, an annotation A is *specific* whenever the root of A , denoted by $\text{root}(A)$, and the leaves of A are *specific* nodes; otherwise, A is *generic*. For example, the annotation trees associated to the documents D_1 and D_2 in figure 2 are specific annotations. By contrast, the common subtree of $A(D_1)$ and $A(D_2)$ given in figure 3 is a generic annotation to which more than one document can be attached, here for example D_1 and D_2 .

Let D_1 and D_2 be two documents, and $A(D_1) = (N_1, E_1)$ and $A(D_2) = (N_2, E_2)$ their respective specific annotation trees. When there is no ambiguity, $A(D_1) = (N_1, E_1)$ and $A(D_2) = (N_2, E_2)$ are written for short respectively A_1 and A_2 . As introduced just above, a *subsumption* (a partial order relation) and an *instantiation* relations are defined on specific and generic annotations (these subsumption and instantiation relations are inspired from *subsumption* of molecular structures detailed in [15]).

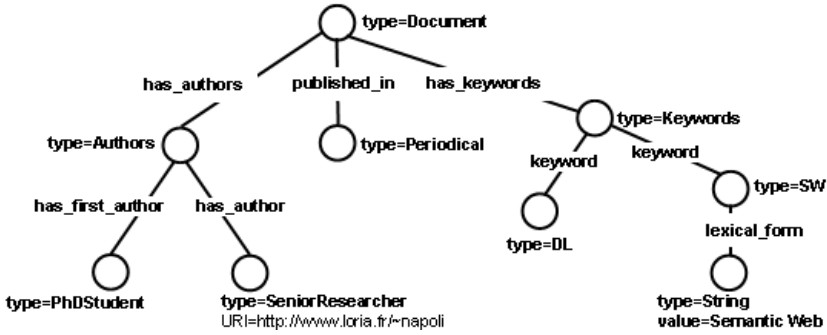


Fig. 3. The subtree generalizing the two annotation trees given in in figure 2



Fig. 4. The links existing between specific annotations, generic annotations and the domain ontology

Definition 2. An annotation $A_1 = (N_1, B_1)$ is subsumed by an annotation $A_2 = (N_2, B_2)$, denoted by $A_1 \sqsubseteq A_2$, if there exists a subtree $A_1' = (N_1', B_1')$ of A_1 , and an isomorphism μ between A_1' and A_2 preserving types, i.e. for each edge $e_2 = (n_2, a_2, n_2')$ in A_2 , there exists an edge $e_1 = (n_1, a_1, n_1')$ in A_1' , where $n_2 = \mu(n_1)$ and $\text{type}(n_1) \sqsubseteq \text{type}(n_2)$, $a_2 = \mu(a_1)$ and $\text{type}(a_1) \sqsubseteq \text{type}(a_2)$, $n_2' = \mu(n_1')$ and $\text{type}(n_1') \sqsubseteq \text{type}(n_2')$.

A specific annotation $A_1 = (N_1, E_1)$ is an instance of a generic annotation $A_2 = (N_2, E_2)$ if and only if $A_1 \sqsubseteq A_2$.

Two remarks have to be made: (i) the most general annotation is supposed to be identified with \top , the root of the hierarchy of concepts and relations, (ii) subsumption on annotations is supposed to hold only between connected annotation trees.

Figure 4 illustrates how generic and specific annotations are related to each other. A specific annotation is associated to a particular document and is composed of several elements of annotation (annotation items) that are typed by classes of the ontology \mathcal{O}_D . Moreover, a specific annotation is an instance of a generic annotation.

The different elements composing semantic annotations may be represented using semantic Web languages such as RDF, RDF-SCHEMA or OWL [8,2]. The annotation trees can be considered either as RDF graphs or XML document models. In this way, the present work can be applied to the comparison of XML

documents (considered as trees). Moreover, based on annotation trees, it becomes possible to compare two documents on the basis of their content, by computing the “semantic similarity” of the two documents. For example, the documents D_1 and D_2 in figure 2 are “semantically similar” because they both describe a document written by an author named *Amedeo Napoli*, in collaboration with another author that is a *PhDStudent*, published in a *periodical*, and concerned with *Description Logics* and *Semantic Web*. Such a comparison is based on a subsumption test of the annotation trees of the considered documents. Moreover, the generalized common subtree of the annotation trees must be *sufficiently sized*, i.e. the size of the subtree must be greater than or equal to a given threshold. These constructions, based on annotation trees and subsumption of annotation trees, are made explicit in the next section.

3 Semantic Similarity and Document Classification

The semantic similarity between two documents is based on the semantic similarity between their annotation trees. The similarity tree shared by two annotation trees is also defined, for being used in a classification process aimed at categorizing Web documents.

3.1 Semantic Similarity and Similarity Tree

First of all, the notion of least common subsumer (LCS) within a class hierarchy is introduced, than can be likened to the LCS operation in description logics [4,5,6] or to the projection in the conceptual graph theory [7,14].

Definition 3. *Given the ontology \mathcal{O}_D and two classes C_1 and C_2 , the class C is the least common subsumer (class) of the classes C_1 and C_2 , denoted by $C = \text{lcs}(C_1, C_2)$, if $C \neq \top$, $C_1 \sqsubseteq C$ and $C_2 \sqsubseteq C$, and whenever $C_1 \sqsubseteq C'$ and $C_2 \sqsubseteq C'$ then $C \sqsubseteq C'$, i.e. C is minimal among the subsumers of the classes C_1 and C_2 .*

Based on this definition, given two annotation trees A_1 and A_2 , two nodes $n_1 \in N_1$ and $n_2 \in N_2$ are said to be *semantically similar* whenever one of the following conditions is verified:

- $\text{type}(n_1) = \text{type}(n_2) = \text{String}$ and $\text{value}(n_1) = \text{value}(n_2)$.
- $\text{type}(n_1) \neq \text{String}$, $\text{type}(n_2) \neq \text{String}$, and $C = \text{lcs}(\text{type}(n_1), \text{type}(n_2))$ exists in \mathcal{O}_D .

By convenience, the class C is said to be the least common subsumer of the two nodes n_1 and n_2 , denoted by $C = \text{lcs}(n_1, n_2)$. In the same way, two edges $e_1 = (n_1, a_1, n'_1)$ and $e_2 = (n_2, a_2, n'_2)$ are *semantically similar* whenever the following is verified: n_1 is semantically similar to n_2 , n'_1 is semantically similar to n'_2 , and the class $A = \text{lcs}(\text{type}(a_1), \text{type}(a_2))$ exists in \mathcal{O}_D . By analogy with nodes, the edge $e = (n, a, n')$ where $n = \text{lcs}(n_1, n_2)$, $n' = \text{lcs}(n'_1, n'_2)$, and $a = \text{lcs}(a_1, a_2)$, is said to be the least common subsumer of the edges $e_1 = (n_1, a_1, n'_1)$ and $e_2 = (n_2, a_2, n'_2)$, denoted by $e = \text{lcs}(e_1, e_2)$.

For example, considering the two annotation trees in figure 2, the following elements of semantic similarity may be extracted:

- The root of A_1 , labeled by `type = HtmlPage`, is similar to the root of A_2 , labeled by `type = BibTexInproceedings`, with `Document = lcs(HtmlPage,-BibTexInproceedings)`.
- The edges labeled in A_1 by `has_author`, `author`, and `has_keywords`, are identical to the corresponding edges in A_2 .
- The edges issued from the roots $\text{root}(A_1)$ and $\text{root}(A_2)$, labeled by `has_series` and `published_in`, are similar to `published_in = lcs(published_in, has_series)`.
- The two leaves labeled by `type = PhDStudent` are similar, even if their values are not equal.

The notion of semantic similarity can be generalized to annotation trees and thus to documents in the following way. Let D_1 and D_2 be two documents, and A_1 and A_2 their respective specific annotation trees. As A_1 and A_2 are specific annotation trees, they have an associated generic annotation tree, namely the generic annotation they are an instance of. Then, the annotation trees A_1 and A_2 are similar if there exists an annotation tree that is the least common subsumer of A_1 and A_2 . For example, the similarity tree of the two annotation trees shown in figure 2 is given in figure 3. More precisely, the semantic similarity for annotation trees is defined as follows.

Definition 4. *The two annotation trees $A_1 = (N_1, E_1)$ associated to the document D_1 , and $A_2 = (N_2, E_2)$ associated to the document D_2 , are said to be semantically similar with degree α , if there exists an annotation tree denoted by $A_S(A_1, A_2) = (N, E)$ that is the least common subsumer of A_1 and A_2 . Moreover, the degree α is given by:*

$$\alpha((A_1, A_2) | A_S(A_1, A_2)) = \frac{|A_S(A_1, A_2)|^2}{|A_1| \cdot |A_2|}$$

where $|A_1|$ denotes the cardinal of the node set of the tree A_1 , and $|A_S(A_1, A_2)|$ the cardinal of the node set of the tree $A_S(A_1, A_2)$.

A constraint can be set up on the similarity degree, such that the degree of the similarity tree $A_S(A_1, A_2)$ of A_1 and A_2 must be greater or equal to a fixed similarity threshold σ_{sim} . For example, the two annotation trees A_1 and A_2 on figure 2 are semantically similar with the similarity tree shown on figure 3, and a degree equal to $\alpha((A_1, A_2) | A_S(A_1, A_2)) = 9^2/(14 \cdot 19) \approx 0.305$. This similarity is acceptable for a threshold σ_{sim} of 0.25 for example.

3.2 An Algorithm for Constructing a Similarity Tree

The semantic similarity between documents depends on their annotation trees, thus on the semantic similarity between nodes and edges of the annotation trees, on a similarity degree, on a similarity threshold σ_{sim} , but also on isomorphism between trees (linked to subsumption between annotations). Indeed, the building

of $A_S(A_1, A_2)$ is based on two *generalizations*, say γ_1 and γ_2 , such that $A_S(A_1, A_2) = \gamma_1(A_1) = \gamma_2(A_2)$ (as in a unification process). Moreover, it has already been remarked that the building of the similarity tree can be likened to the building of the least common subsumers in description logics [4,5,6], or to the projection of conceptual graphs [7,14].

The building of a similarity tree, given two annotation trees A_1 and A_2 , may be achieved according to the following rules, the generalizations γ_1 and γ_2 being based on the definitions of the semantic similarity between nodes and edges.

- When two nodes $n_1 \in N_1$ and $n_2 \in N_2$ are semantically similar, then the node n in $A_S(A_1, A_2)$ results from the generalization of n_1 and n_2 , with $n = \gamma_1(n_1) = \gamma_2(n_2)$.
Practically, a function **CreateNode**(n_1, n_2) takes the nodes $n_1 \in N_1$ and $n_2 \in N_2$ as inputs, and searches for the class $n = \text{lcs}(n_1, n_2)$ in \mathcal{O}_D , and then builds the node labeled by n . In the case $n_1 = n_2$, with n_1 and n_2 having the same value or URI, this value or URI is attached to the node labeled by n .
- When a node $n_1 \in N_1$ does not have any similar node in N_2 , then the substitution cannot be computed and the node n_1 is not taken into account in the building of the similarity tree. For example, this is the case of the node **type** = **AssistantProfessor**, **URI** = **http://.../lieber** in figure 2, for A_2 .
- When two edges $e_1 = (n_1, a_1, n'_1) \in E_1$ and $e_2 = (n_2, a_2, n'_2) \in E_2$ are semantically similar, then the edge $e = (n, a, n')$ in $A_S(A_1, A_2)$ is obtained by generalizing e_1 and e_2 , with $n = \gamma_1(n_1) = \gamma_2(n_2)$, $a = \gamma_1(a_1) = \gamma_2(a_2)$, and $n' = \gamma_1(n'_1) = \gamma_2(n'_2)$.
Practically, a function **CreateEdge**(e_1, e_2) takes the two edges e_1 and e_2 as inputs, and searches for the edge $e = \text{lcs}(e_1, e_2) = (n, a, n')$ in \mathcal{O}_D .
- When an edge $e_1 \in E_1$ does not have any similar edge in E_2 , then the substitution is not defined and the edge e_1 is not taken into account in the building of the similarity tree.

The similarity tree is built by searching, starting from the roots $\text{root}(A_1)$ and $\text{root}(A_2)$, the largest subtree that can be matched according to the previous rules. The algorithm produces as output the correspondence between the nodes of A_1 and the nodes of A_2 :

1. The set of node correspondence between A_1 and A_2 for semantically similar nodes, denoted by $\Pi_{\text{node}}(A_1, A_2)$. For example, the correspondence for the annotation trees given in figure 2 is:
 $\Pi_{\text{node}}(A_1, A_2) = \{(a, r), (b, s), (c, t), (d, u), (f, v), (g, w), (h, z), (i, x), (j, y)\}$
2. The set of edge correspondence between A_1 and A_2 for semantically similar edges, is denoted by $\Pi_{\text{edge}}(A_1, A_2)$. For example, $\Pi_{\text{edge}}(A_1, A_2)$ contains the pairs of edges associated with the correspondence $\Pi_{\text{node}}(A_1, A_2)$.
3. The set of nodes of A_1 and the set of nodes of A_2 that have not to be taken into account. For example, all nodes of A_1 and A_2 that are not in $\Pi_{\text{node}}(A_1, A_2)$ are not taken into account for the construction of the similarity tree $A_S(A_1, A_2)$.

Two remarks have to be done. Firstly, the generalizations γ_1 and γ_2 are different only when the nodes n_1 or n_2 , or the edges e_1 or e_2 , do not have any similar node or edge. Secondly, the **lcs** operation here is much more simpler than the

general building of a LCS in descriptions logics (see [5]): the LCS is actually not built but *searched for* within the $\mathcal{O}_{\mathcal{D}}$ concept hierarchy, and thus corresponds to an already existing concept.

3.3 An Algorithm for the Classification of Annotation Trees

Let D_1 and D_2 be two semantically similar documents with the similarity tree $A_S(A_1, A_2) = (N, E)$, built from the annotation trees $A_1 = A(D_1) = (N_1, E_1)$ and $A_2 = A(D_2) = (N_2, E_2)$. A classification algorithm may be proposed, for retrieving the *best instantiation class* with respect to the similarity degree for the annotation tree associated to a given document.

Let DOCS be a set of documents, $\mathcal{O}_{\mathcal{D}}$ the ontology associated to DOCS, and $\mathcal{H}_{\mathcal{A}}$ a hierarchy of annotation trees associated to DOCS (see figure 5). The classes in $\mathcal{H}_{\mathcal{A}}$ represent generic annotations related to sets of documents in DOCS, with respect to the ontology $\mathcal{O}_{\mathcal{D}}$. Actually, the $\mathcal{H}_{\mathcal{A}}$ hierarchy may also be considered either as an ontology of generic annotations related to the documents in DOCS, or to a classification of the documents in DOCS.

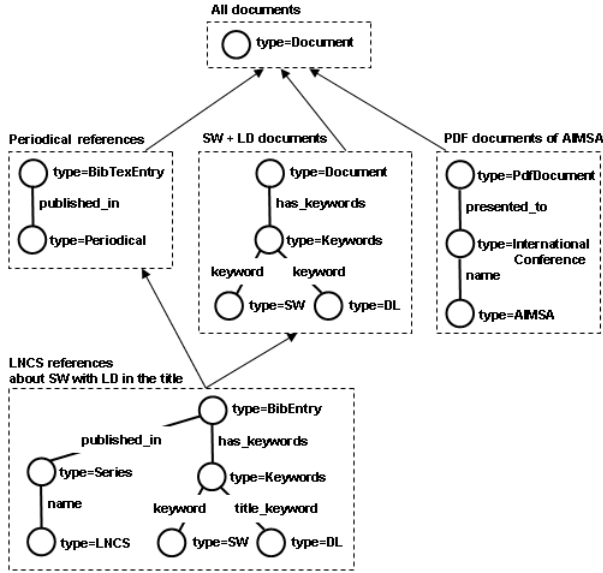


Fig. 5. A hierarchy $\mathcal{H}_{\mathcal{A}}$ of annotations

Given the ontology $\mathcal{O}_{\mathcal{D}}$ and an annotation hierarchy $\mathcal{H}_{\mathcal{A}}$, the principle of the classification algorithm is the following. The classes subsuming $A_1 = A(D_1) = (N_1, E_1)$ are searched in $\mathcal{H}_{\mathcal{A}}$ using a depth-first search. More precisely, there is a search for the classes C in $\mathcal{H}_{\mathcal{A}}$ subsuming the generic annotation associated to A_1 , i.e. A_1 is an instance of C (as stated in definition 2). Zero or more subsuming classes may exist in $\mathcal{H}_{\mathcal{A}}$:

- When no subsuming class exists, then the type of the document D_1 is not represented in \mathcal{H}_A . A new class representing D_1 can be created, as in an incremental classification algorithm [11]. Maybe, this means that the themes of the document D_1 are out of the scope of those underlying the ontology \mathcal{O}_D and the documents in DOCS.
- If one or more classes subsume A_1 , then similar documents to D_1 are found, namely the documents that are instances of classes subsuming A_1 or the subsumers of A_1 . The subsumers can be sorted according to their similarity degree, i.e. more a class is close to D_1 better it is ranked.

For example, the annotation tree A_2 associated to the document D_2 is classified in the hierarchy \mathcal{H}_A as follows (cf. figure 5). According to the ontology \mathcal{O}_D and to the hierarchy \mathcal{H}_A , the class ‘**Periodical references**’ may be selected (assimilating the series ‘**LNCS**’ to a periodical publication), and then the class ‘**LNCS references about SW with DL in the title**’ is also selected, as the most specific subsumer: both classes subsume the annotation tree A_2 . Then, the class ‘**SW + DL documents**’ is selected, but the class ‘**PDF documents of AIMSA**’ is discarded. Documents similar to D_2 have been found, namely the instances of the classes ‘**Periodical references**’ and ‘**LNCS references about SW with DL in the title**’.

Such an algorithm can be used for classifying documents according to their content, for comparing documents and for finding similar documents. Moreover, the classification algorithm can be used for extending the \mathcal{H}_A hierarchy, with respect to the \mathcal{O}_D ontology, by proposing new annotations classes. Actually, these aspects of the present work are research perspectives and a straightforward continuation of this work.

In addition, a *semantic similarity measure* between the two documents D_1 and D_2 may be defined, taking into account the proportion of semantically similar nodes in A_1 and A_2 with respect to $A_S(A_1, A_2)$, and the similarities between the types, URI and values of the leaves in A_1 and A_2 . A first formalization is proposed in [1], and the definition of a more acceptable and efficient form is currently under investigation.

4 Discussion and Conclusion

The present approach proposed for comparing documents represented by annotation trees shows a number of advantages. On the one hand, relations existing between terms describing the content of documents can be taken into account and used for having a better understanding of the documents, i.e. for knowing whether a document D_1 is more general than a document D_2 , with respect to the ontology \mathcal{O}_D and to the content of documents. On the other hand, annotation trees can be manipulated, matched, and compared, using standard tools adapted to XML document manipulation. Moreover, the present approach takes explicitly advantage of the domain ontology \mathcal{O}_D , allowing sophisticated reasoning, e.g. for finding analog documents, where two documents D_1 and D_2 are considered as analogs when there exists a “similarity path”, i.e. a sequence of similar documents, between their annotation trees (see for example [12]).

This approach has been implemented, and an experiment has been carried out on bibliographic documents, for comparing the behavior of the present approach and more classical information retrieval approaches, based on vectors and a thesaurus (organized by a generic/specific relation). The annotations are composed of relations such as **written-by** (an author), **published-in** (a year), **edited-in** (publication-support), **talking-about** (a keyword), etc. Some documents are found to be similar with the vector model, while they are found to be not similar according to the annotation tree approach. For example, two references containing the same type of authors, or published in a same type of publication support, are found to be similar in the annotation tree approach, while they are not (because of the need of exact matching) in the vector approach. It would be interesting then to combine both approaches, the robust and efficient methods based on keyword vectors, and the approach based on the annotation trees and the domain ontology. Moreover, the ontology that has been used for the experiment remains to be improved, with more complex concept and relation descriptions.

Finally, it must be remarked that the present approach, based on annotation trees, may be well-suited for: (i) the detection or search of resources represented as RDF graphs (a format that is in accordance with annotation trees), (ii) the more general task of retrieval of XML documents according to a set of constraints, because of the tree-based representation that can be associated to an XML document.

References

1. R. Al-Hulou, A. Napoli, and E. Nauer. Une mesure de similarité sémantique pour raisonner sur des documents. In J. Euzenat and B. Carré, editors, *Langages et modèles à objets, Lille (LMO'04)*, pages 217–230. Hermès, L'objet 10(2–3), 2004.
2. G. Antoniou and F. van Harmelen. *A Semantic Web Primer*. The MIT Press, Cambridge, Massachusetts, 2004.
3. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, Cambridge, UK, 2003.
4. F. Baader and B. Sertkaya. Applying formal concept analysis to description logics. In P. Eklund, editor, *Second International Conference on Formal Concept Analysis, Sydney (ICFCA 2004)*, Lecture Notes in Artificial Intelligence 2961, pages 261–286. Springer, Berlin, 2004.
5. F. Baader, B. Sertkaya, and A.-Y. Turhan. Computing the least common subsumer w.r.t. a background terminology. In J.J. Alferes and J.A. Leite, editors, *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA 2004), Lisbon, Portugal*, volume 3229 of *Lecture Notes in Computer Science*, pages 400–412. Springer-Verlag, 2004.
6. Franz Baader. Computing the least common subsumer in the description logic \mathcal{EL} w.r.t. terminological cycles with descriptive semantics. In *Proceedings of the 11th International Conference on Conceptual Structures, ICCS 2003*, volume 2746 of *Lecture Notes in Artificial Intelligence*, pages 117–130. Springer-Verlag, 2003.
7. M. Chein and M.-L. Mugnier. Conceptual graphs: Fundamental notions. *Revue d'intelligence artificielle*, 6(4):365–406, 1992.

8. D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster, editors. *Spinning the Semantic Web*. The MIT Press, Cambridge, Massachusetts, 2003.
9. S. Handschuh and S. Staab, editors. *Annotation for the Semantic Web*. Volume 96 Frontiers in Artificial Intelligence and Applications. IOS Press, Amsterdam, 2003.
10. J. Heflin, J.A. Hendler, and S. Luke. SHOE: A blueprint for the semantic web. In *Spinning the Semantic Web*, pages 29–63, 2003.
11. P. Langley. *Elements of Machine Learning*. Morgan Kaufmann Publishers, San Francisco, California, 1996.
12. J. Lieber and A. Napoli. Correct and Complete Retrieval for Case-Based Problem-Solving. In H. Prade, editor, *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI'98), Brighton, UK*, pages 68–72. John Wiley & Sons Ltd, Chichester, 1998.
13. A. Maedche, S. Staab, N. Stojanovic, R. Studer, and Y. Sure. SEmantic portAL: the SEAL Approach. In D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster, editors, *Spinning the Semantic Web*, pages 317–359. The MIT Press, Cambridge, Massachusetts, 2003.
14. M.L. Mugnier. On generalization/specialization for conceptual graphs. *Journal of Experimental & Theoretical Artificial Intelligence*, 6(3):325–344, 1995.
15. A. Napoli, C. Laureço, and R. Ducournau. An object-based representation system for organic synthesis planning. *International Journal of Human-Computer Studies*, 41(1/2):5–32, 1994.
16. F. Sebastiani, editor. *Advances in Information Retrieval, 25th European Conference on IR Research, ECIR 2003, Pisa, Italy, April 14-16, 2003, Proceedings*, Lecture Notes in Computer Science 2633. Springer, 2003.
17. S. Staab and R. Studer, editors. *Handbook on Ontologies*. Springer, Berlin, 2004.
18. V. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E. Motta, and F. Ciravegna. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Journal of Web Semantics*, 4(1), 2005.

Cooking an Ontology^{*}

Ricardo Ribeiro², Fernando Batista², Joana Paulo Pardal¹,
Nuno J. Mamede¹, and H. Sofia Pinto¹

¹ IST/INESC-ID Lisboa

² ISCTE/INESC-ID Lisboa

Rua Alves Redol, 9, 1000-029 Lisboa, Portugal

{rdmr, fmmmb, joana, njm}@l2f.inesc-id.pt,
sofia@vinci.inesc-id.pt

Abstract. An effective solution to the problem of extending a dialogue system to new knowledge domains requires a clear separation between the knowledge and the system: as ontologies are used to conceptualize information, they can be used as a means to improve the separation between the dialogue system and the domain information. This paper presents the development of an ontology for the cooking domain, to be integrated in a dialog system. The ontology comprehends four main modules covering the key concepts of the cooking domain – actions, food, recipes, and utensils – and three auxiliary modules – units and measures, equivalencies and plate types.

Keywords: ontology construction, knowledge representation, dialogue systems, natural language processing.

1 Introduction

An effective solution to the problem of extending a dialogue system to new knowledge domains requires a clear separation between the knowledge and the system: as ontologies are used to conceptualize information, they can be used as a means to improve the separation between the dialogue system and the domain information. Having a generic spoken dialogue system able to manage specific devices at home, such as TVs, lamps and windows, a natural step was to extend it to other domains. The cooking domain appeared as an interesting application area since the dialog system was being used in an exhibition called "The House of the Future". At that point we had an electronic agent that could answer to voice commands allowing easier control of home devices using only voice. Ontologies are used to conceptualize knowledge. So, if we managed to "teach" the system how to use the ontological knowledge, we would increase independence easing the task of adding new domains, since this would be as easy as plugging an appropriate ontology.

The paper is structured as follows: section 2 presents a brief state-of-the-art in ontology engineering; section 3 presents previously developed ontologies that could be used if adequate; section 4 presents our ontology; section 5 details its development process; section 6 describes the work done so far in integrating the ontology in the dialogue system; future work and conclusions complete the paper.

^{*} This paper has been partially supported by Departamento de Ciências e Tecnologias de Informação – ISCTE and by FCT project DIGA – POSI/PLP/41319/2001.

2 State of the Art

The work on the ontology field goes back to the beginning of 1990. The first ontologies were built from scratch and made available in order to demonstrate their usefulness. By that time no methodologies or guidelines were available to guide or ease the building process. After some experiences on the field, [1] introduced some principles for the design of ontologies. Gruber's work was the first to describe the role of ontologies in supporting knowledge sharing activities, and presented a set of guidelines for the development of ontologies. The ontology building process became clearer, with the continuous development of several other ontologies. As a consequence, the first methodologies for building ontologies were proposed in 1995, leading to the emergence of the ontological engineering field.

According to [2], three different generations of methodologies can be distinguished. The first generation corresponds to the first attempts on understanding how ontologies could be built. The building process was the main issue, postponing problems, such as maintenance and reuse. Methodologies used in TOVE [3] and ENTERPRISE [4] fit in this first generation. The second generation considers performing specification, conceptualization, integration, and implementation as often as required, during the ontology lifetime. The initial version of METHONTOLOGY [5] belongs to the second generation. The current version of the METHONTOLOGY and OTK [6] can be included in this last generation of methodologies, where topics such as *reusability* and *configuration management*, became activities of the development process. Currently, neither a standard methodology exists, nor a sufficiently mature one was found having a considerable user community.

Recent years have seen a surge of interest in the discovery and automatic creation of complex, multi-relational knowledge structures, as several workshops on the field illustrate. For example, the natural language community is trying to acquire word semantics from natural language texts. A remaining challenge is to evaluate in a quantitative manner how useful or accurate the extracted ontology classes, properties and instances are. This is a central issue as it is currently very hard to compare methods and approaches, due to the lack of a shared understanding of the task at hand and its appropriate metrics.

3 Related Ontologies

The motto *ontologies are built to be reused* [5] conveys in an appropriate manner the ideas originally proposed by [1]. Therefore, the first step was to survey existing knowledge sources on the cooking domain and check their adequacy. Of these sources: (a) USDA National Nutrient Database for Standard Reference is a database made by the United States Department of Agriculture to be the major source of food composition data in the United States. In its 18th release (SR18) comprehends 7,146 food items and up to 136 food components [7]; (b) AGROVOC is a multi-lingual thesaurus made by the Food and Agriculture Organization of the United Nations (FAO) that has about 17,000 concepts and 3 types of relations (preferred term, related term and broader term) [8]; (c) [9] presents the development of a wine (main focus), food and appropriate combinations of wine with meals ontology; (d) [10] presents a specialized wine ontology

that covers maceration, fermentation processes, grape maturity state, wine characteristics, and several classifications according to country and region where the wine was produced; (e) [11] describes an ontology of culinary recipes, developed to be used in a semantic querying system for the Web.

These ontologies did not cover what was intended in our project: some were too specific, focusing on issues like wine (c and d) or nutrients themselves (a), others not deep enough (e), focused (as stated in their objectives) in building a classification – adequate to a specific application – of part of the knowledge we intended to structure.

4 Cooking Ontology

The development of the cooking ontology did not follow a specific ontology development methodology, but was strongly influenced by the ideas presented in [12].

[recipes]
How do I make recipe <i>R1</i> ?
What are the quantities to use when making recipe <i>R1</i> for 4 persons?
[actions]
How do I do <i>A1</i> ?
[times]
Which are the recipes that take less than 10 minutes to make?
[food]
Which recipes have food item <i>F1</i> , but not <i>F2</i> ?
Which are the recipes that have as main ingredient food item <i>F1</i> ?
[utensils]
Which utensils are used in recipe <i>R1</i> ?
Which recipes can be made using the microwave?
[equivalencies]
How many liters is a cup?

Fig. 1. Competency questions

In brainstorm meetings the comprehension of the domain evolved to the identification of four key areas of the cooking domain. As this areas are wide and independent enough they were split into modules: (i) actions; (ii) food; (iii) recipes; (iv) kitchen utensils. Also three auxiliary modules were found: units and measures, equivalencies, and plate types. To define the scope of the ontology, informal competency questions were formulated, figure 1. These questions addressed specifically each of the previously identified areas and guided the building process. Their satisfaction was continuously evaluated: whenever a new release was made available it was checked if they were being correctly answered.

Figure 2 shows the relations between the main concepts. A Recipe is organized into three Phases: *preparation*, *cooking* and *presentation*. Each Phase is an ordered sequence of Tasks, some of which may be optional. A Task is composed by an Action, its duration

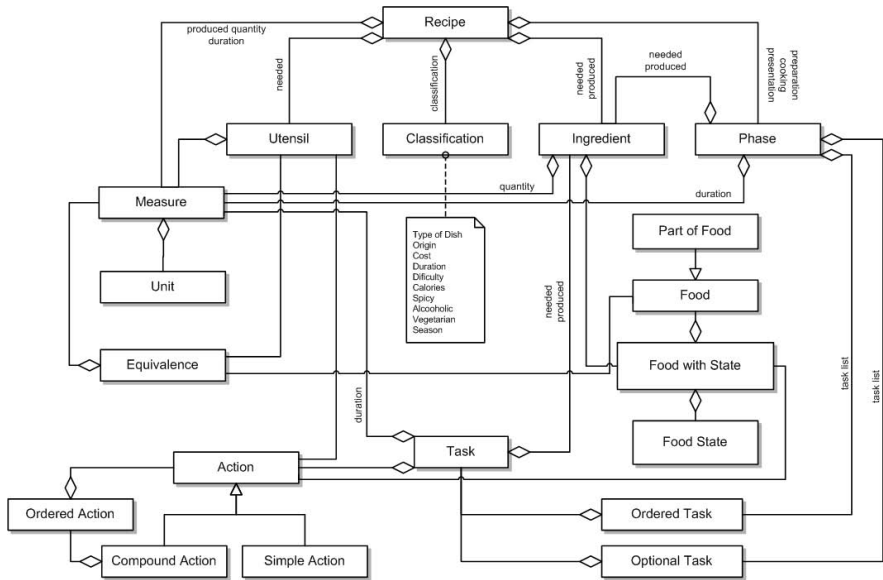


Fig. 2. Main concepts

time, and incorporates information about *needed* and *produced* Ingredients. Some of the Ingredients of the Tasks are also part of the Phase that uses them (intermediate result Ingredients are not accounted in the Phase concept). The Phase also has a duration (Measure of time Units). Each Recipe has a Classification, a list of Ingredients and required Utensils.

The current version of the cooking ontology has about 1151 classes, 92 slots (of which 52 establish relations between classes) and 311 instances, distributed by the seven modules.

5 Building Process

When building the ontology, activities like brainstorm sessions, knowledge validation and disambiguation, conceptualization and formalization, or evaluation, among others, were done along the project by all team members in weekly meetings. Acquisition, conceptualization, and formalization of specific knowledge was divided into the identified areas and assigned to different team members. The knowledge model was formalized using Protégé [13], which can also be used to automatically generate the ontology code.

The requirements specification was defined from the dialogue system context, the earlier phases of knowledge acquisition, and the competency questions.

5.1 Knowledge Acquisition

Knowledge acquisition, the first step of the building process, began with reading and selecting available cooking books. The knowledge sources had different views about the

subject, but they all agreed on the separation of concepts. For example, almost every source had a description of the kitchen tools; animal (cow, pork, rabbit, etc.) parts; and fish types. The first step to organize concepts was the knowledge in these sources. Recipes were viewed as algorithms used to define a set of basic actions that later were used to describe Recipe concepts.

5.2 Conceptualization

The main activities in conceptualization were (i) identification of concepts and their properties; (ii) classification of groups of concepts in classification trees; (iii) description of properties; (iv) identification of instances; (v) description of instances. During this phase, discussion and validation sessions were also held to identify the relations between classification trees; initial debates were held to discuss how concepts should be modelled (classes versus instances); and, harmonization of the identified properties (within the several composing modules) and their definitions was performed.

5.3 Formalization

One of the main issues in formalization concerned relations between concepts. As it was described before, several concepts (for example, Food and Utensils) entail own hierarchies. Concepts within these hierarchies were associated through IS-A relations. Attribute-based relations were used to associate concepts from the several hierarchies and the other concepts (such as Task and Recipe). For example, a recipe uses utensils and that is stated as a slot in the Recipe class.

Another key issue in formalization was to decide if each concept should be formalized as a class or instance. Several discussions and experiments were needed to understand the best way to formalize the concepts and their relations. Here the main complexity arose from the needed global coherence as the knowledge formalization must be capable of describing a Recipe and all its related information.

Some concepts were formalized as classes and their instances use the defined hierarchies as taxonomies (the values of the attributes are of class type). For example, a Recipe has several attributes of this kind.

Food and Utensil concepts were formalized as classes. Food (abstract) classes will have no instances as they are used as a taxonomy to characterize Ingredient instances. Utensil instances depend on the usage of the ontology. For example, considering the context of a dialog system (like the one described earlier), Utensil instances would be the real utensils that the user can operate. Actions were formalized using classes and instances: classes to arrange the hierarchy and instances to describe the leaves of the classification tree.

5.4 Evaluation

Two types of evaluation were performed, neither using a standard methodology like On-toClean [14], nor [15]: an internal evaluation performed by the whole team during the ontology life cycle, and an external evaluation performed by the client. The client supervised the releases mainly by asking the defined competency questions and checking

whether the ontology could answer them. Since no inference is available at the moment, all verifications were done by checking whether the information was available and if the right relations existed. In later stages, this checking can be done automatically by using an inference engine.

6 Integration in a Dialogue System

Some work has already been done that showed the advantages of using ontologies to enrich spoken dialogue systems with domain knowledge [16,17].

As the motivation for this project was extending an existing spoken dialogue system [18], the next step is to use the resulting ontology to enrich it.

The current version of the ontology is useful and usable in that context. Some preliminary tests have been made to allow the autonomous agent to start helping in kitchen

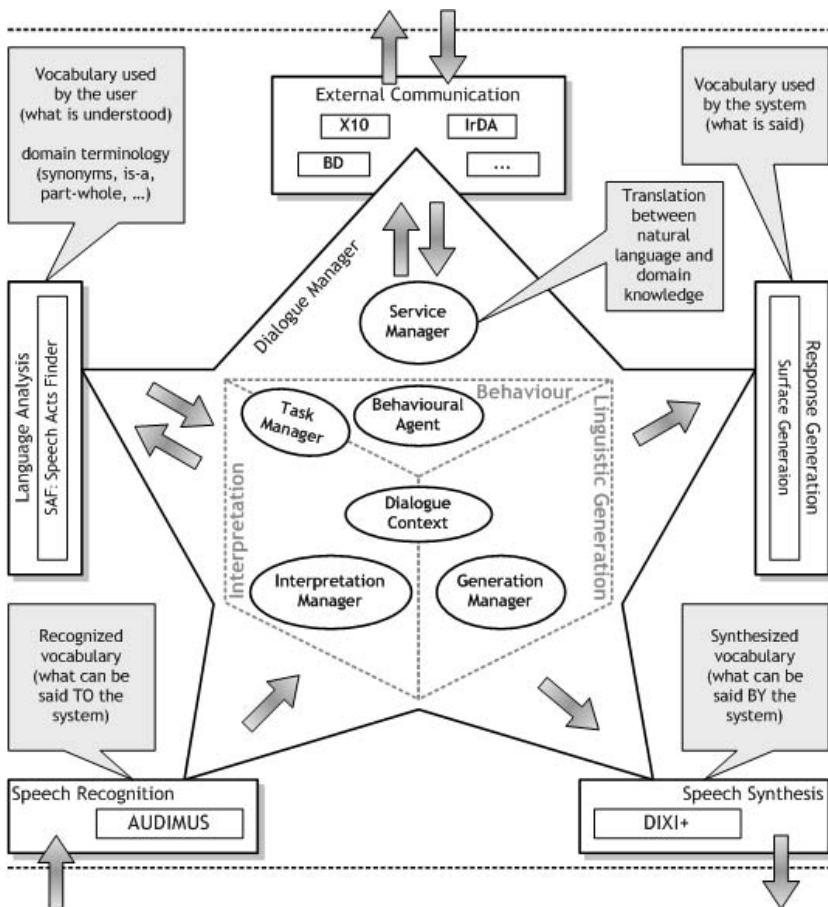


Fig. 3. STAR architecture

tasks. The current version of the system takes a list of recipes, asks the user which one he wants to hear and reads it to the user.

Figure 3 shows the main architecture of our system. The system has six main modules: speech recognition, language analysis, external communication, response generation, speech synthesis and – at the centre, communicating with all the others – a dialogue manager. For each of the modules, a gray box shows the domain knowledge that can be used to produce better results. In the middle, the components of the dialogue manager are presented: interpretation manager, dialogue context, task manager, behavioural agent, service manager and generation manager. These modules are responsible for the interpretation, the behaviour and the linguistic generation.

When the user says something, speech recognition needs to know the words related to the domain to add them to the language model. After the speech is transformed into text, the meaning of what was said depends on the dialogue context (what has been said before) and on the knowledge domain. For instance, the taxonomical knowledge allows a smarter reasoning and a better dialogue sequence. After understanding what has been said, it is necessary to translate that into the external systems language. Once again, this knowledge can be arranged in an ontology. After the execution of the user's commands, a response is produced to inform on the results. The naturality of the system depends on the chosen vocabulary. For instance, the usage of synonyms in the communication. Also when there is the need of some clarification, the sequence of the questions can be enhanced if the questions are produced ordered by their relatedness.

The ontology gathers all the knowledge that currently is spread through the modules of the system. This is an advantage as all the knowledge information will be concentrated in the same module, the ontology. Presently, it is already necessary to collect the knowledge when integrating a new domain. Using an ontology, instead of splitting that knowledge into the relevant modules, turns it pluggable (plug-and-play).

Our dialogue system has an architecture similar to the well known used by TRIPS [19]. It will be interesting to explore how the knowledge stored in an ontology can be used automatically and dynamically by a dialogue system. For example, the words that name the concepts can be used to collect the related vocabulary. The usage of a spread architecture eases the transference of this technique to similar systems.

7 Future Work

Apart from adding new concepts, sharing, reusing, maintaining and evolving, which are important issues for the future, each module has its own characteristics that can be improved.

In the food module, an interesting possibility is to add new food classifications based on different criteria. For example, a flavour based classification or one based on the nutrition pyramid would increment the information level about food items. The observation of a large set of important characteristics of kitchen utensils suggests that additional information can be added to each concept, either in the scope of the documentation, or in the set of defined properties. An improvement to the actions module could be the integration of a process ontology to define complex actions and recipes. Such restructuring should be carefully thought, since the benefits may be outweighed by the difficulties.

In the future, when using the ontology in our Dialogue System, the application could include a personalized configuration to specify the real utensils that the user has at home as Utensils instances. In that case, when referring to the objects the system could even refer the place, for example the drawer, where they are stored. User adaptation could focus on issues – with different impacts on the work done – like the following: kind of dish could be extended to take some cultural differences into consideration – *Pasta* is eaten before the main dish (as an appetizer) by Italians while Portuguese people eat it as a main dish or even as companion for the meat or fish –; the origin of plates could be connected to a Geographical Ontology in order to allow inference on geographical proximity; and, the Season of the year when a meal is more adequate could lead to a new module to be used replacing the current discrete values.

8 Conclusions

The aim for this work consisted on developing an ontology on the cooking domain, in order to be integrated in a dialog system. The resulting ontology covers four main areas of the domain knowledge: food, kitchen utensils, actions and recipes. Food, utensils and actions areas of knowledge are formalized as class hierarchies with instances (in what concerns actions), covering in a considerable extent – at least, accordingly to the used information sources – the target domain. Recipes concepts interconnect concepts from all the other areas, in order to define an adequate model of the cooking domain. Two instances of Recipe were created to demonstrate the usability of the developed specification.

The ontology building process was strongly influenced by METHONTOLOGY and the phases of specification, knowledge acquisition, conceptualization, implementation and evaluation were essential to achieve the intended result.

Despite the problems found, the ontology reached a usable state. All concepts were structured and well documented. The integration with the dialogue system is work in progress and only preliminary tests were conducted, since that effort is part of an on going PhD thesis project.

References

1. Gruber, T.R.: A translation approach to portable ontology specifications. *Knowledge Acquisition* **5**(2) (1993)
2. Pinto, H.S., Martins, J.P.: Ontologies: How can They be Built? *Knowledge Information Systems* **6**(4) (2004)
3. Grüninger, M., Fox, M.: Methodology for the Design and Evaluation of Ontologies. In: *IJCAI'95, Workshop on Basic Ontological Issues in Knowledge Sharing*. (1995)
4. Uschold, M., King, M., Moralee, S., Zorgios, Y.: The Enterprise Ontology. *The Knowledge Engineering Review* **13** (1995) Special Issue on Putting Ontologies to Use.
5. Fernández, M., Gomez-Perez, A., Juristo, N.: METHONTOLOGY: from Ontological Art towards Ontological Engineering. In: *Proc. of the AAAI97 Spring Symposium Series on Ontological Engineering*. (1997)
6. Sure, Y.: Methodology, tools and case studies for ontology based knowledge management. PhD thesis, Universität Karlsruhe (2003)

7. USDA: National Nutrient Database for Standard Reference. www.nal.usda.gov/ (2005)
8. FAO: AGROVOC. www.fao.org/agrovoc/ (2004)
9. Noy, N.F., McGuinness, D.L.: Ontology Development 101: A Guide to Creating Your First Ontology. Technical Report KSL-01-05/SMI-2001-0880, Stanford Knowledge Systems Laboratory/Stanford Medical Informatics (2001)
10. Graça, J., Mourão, M., Anunciação, O., Monteiro, P., Pinto, H.S., Loureiro, V.: Ontology building process: the wine domain. In: Proc. of the 5th Conf. of EFITA. (2005)
11. Villarías, L.G.: Ontology-based semantic querying of the web with respect to food recipes. Master's thesis, Technical University of Denmark (2004)
12. López, M.F., Gómez-Pérez, A., Sierra, J.P., Sierra, A.P.: Building a Chemical Ontology Using Methontology and the Ontology Design Environment. *IEEE Intell. Sys.* **14**(1) (1999)
13. Gennari, J., Musen, M., Ferguson, R., Grosso, W., Crubézy, M., Eriksson, H., Noy, N.F., Tu, S.: The evolution of Protégé: an environment for knowledge-based systems development. *Intl. Journal of Human-Computer Studies* **58**(1) (2003)
14. Guarino, N., Welty, C.: An Overview of OntoClean. In: *Handbook on Ontologies*, Springer-Verlag (2004)
15. Gómez-Pérez, A.: Evaluation of taxonomic knowledge in ontologies and knowledge bases. In: *Banff Knowledge Acquisition for Knowledge-Based Systems, KAW'99. Volume 2.*, University of Calgary, Alberta, Canada (1999) 6.1.1–6.1.18
16. Milward, D., Beveridge, M.: Ontology-based Dialogue Systems. In: *IJCAI'03, Workshop on Knowledge and Reasoning in Practical Dialogue Systems.* (2003)
17. Flycht-Eriksson, A.: Design and Use of Ontologies in Information-providing Dialogue Systems. PhD thesis, School of Engineering at Linköping University (2004)
18. Mourão, M., Madeira, P., Mamede, N.: Interpretations and Discourse Obligations in a Dialog System. In: Proc. of the 6th Intl. Workshop on Computational Processing of the Portuguese Language. Number 2721 in LNAI, Springer-Verlag (2003)
19. Allen, J., Ferguson, G., Swift, M., Stent, A., Stoness, S., Galescu, L., Chambers, N., Campana, E., Aist, G.: Two diverse systems built using generic components for spoken dialogue (recent progress on TRIPS). In: Proc. of the Interactive Poster and Demonstration Sessions at the 43rd Annual Meeting of the ACL. (2005)

Methodology for Bootstrapping Relation Extraction for the Semantic Web

Maria Tchalakova^{1,2}, Borislav Popov¹, and Milena Yankova¹

¹ Ontotext Lab, Sirma Group Corp., 135 Tsarigradsko Chaussee,
Sofia 1784, Bulgaria
<http://www.ontotext.com>

² University of Tübingen, Seminar für Sprachwissenschaft,
Wilhelmstr. 19, 72074 Tübingen, Germany
maria.tchalakova@gmail.com,
{borislav, milena}@sirma.bg

Abstract. The paper describes a methodology for bootstrapping relation extraction from unstructured text in the context of GATE, but also applied to the KIM semantic annotation platform. The focus is on identifying a set of relations between entities previously found by named entity recognizer. The methodology is developed and applied to three kinds of relations and evaluated both with the ANNIE system and the default information extraction module of KIM. The methodology covers the problem of identifying the task, the target domain, the development of training and testing corpora, and useful lexical resources, the choice of a particular relation extraction approach. The application of information extraction for the Semantic Web also brings a new interesting dimension of not merely recognizing the entity type, but going into instantiation of entity references and linking them to an entity instance in a semantic repository.

Keywords: Relation Information Extraction, Semantic Web, Methodology.

1 Rationale

The world has been flooded with information by the phenomenon of the Internet in the recent decade or so. The initial model of the WWW has focused on mark-up languages defining the presentation, but did not address the structural or semantic aspects of the online content. Decades ago, the vision of a network of resources enriched with a machine readable semantic layer have been considered by some individuals (e.g. Jerry Hobbs with his students organized “the summer of semantics” in the late 70s and got some funding on creating an intelligent web. After some time the funding seized, obviously because the ideas were decades ahead of the main stream). However, only in the recent years the idea of the Semantic Web has appeared as a resurrection of this vision and is currently gaining momentum. The current problem is that there is a very limited amount of semantic descriptions on the Web given the scale of the latter. A solution would be to create technologies that generate descriptive metadata automatically. The KIM Platform¹ is such a system for automatic

¹ KIM Platform, <http://www.ontotext.com/kim/>

meta-data generation. It performs shallow text analysis and generates semantic annotations, given that most of the available content on the Web is (at least partly) textually presented. The technique employed is Information Extraction (IE) with emphasis on Named Entities (NE). These entities are a significant hint to the “meaning” of the content, but much more can be done.

In this context, it is very helpful to have a methodology for extending the IE from NE recognition to Relation Extraction (RE), e.g. finding person’s position in an organization, or organization’s location of activity. This motivated the development of RE for the KIM Platform as a methodology as well as concrete implementation for three kinds of relations.

2 Introduction

This paper describes a methodology for bootstrapping relation extraction from unstructured text in the context of the GATE platform², defining the steps needed for a successful and robust RE development. The results of the approach were also integrated in KIM. The work was focused on the default IE module of KIM which was extended toward the extraction of relations between already recognised entities. Since the development of a methodology without a proof-of-concept application is like writing books that nobody reads, the paper also describes the development of a RE module focused on three relations: *Person has Position within Organization*, *Organization activeIn Location*, and *Person has Position within Location*.

The different steps of the methodology are presented in section 3, the techniques used for the relation extraction in the example application – in section 4, and the evaluation results of the example application – in section 5. Some future work is proposed in section 6.

3 Different Steps in the Methodology

In this section the steps in the proposed RE bootstrapping methodology are presented. On Fig. 1 the different phases (influenced to a certain extend by the rule-based approach used in the example application) are depicted. The solid arrows represent the sequence of the phases and the dotted arrows represent the loops in the methodology.

3.1 Task Definition

The first step in the development of a RE module is oriented toward identifying a need of particular relations to be automatically extracted. One should also consider the availability of Named Entity Recognition (NER) modules for recognising those entities that take part in the relations of interest, in order to be able to count on such a module as a basis of the RE development. Thus, for the identification of the task the particular need for extracting relations should first be considered, as well as the availability of the necessary prerequisites, and then its scope and relations in focus should be defined. The interesting aspect of the relationships between real-world

² GATE, <http://gate.ac.uk/>

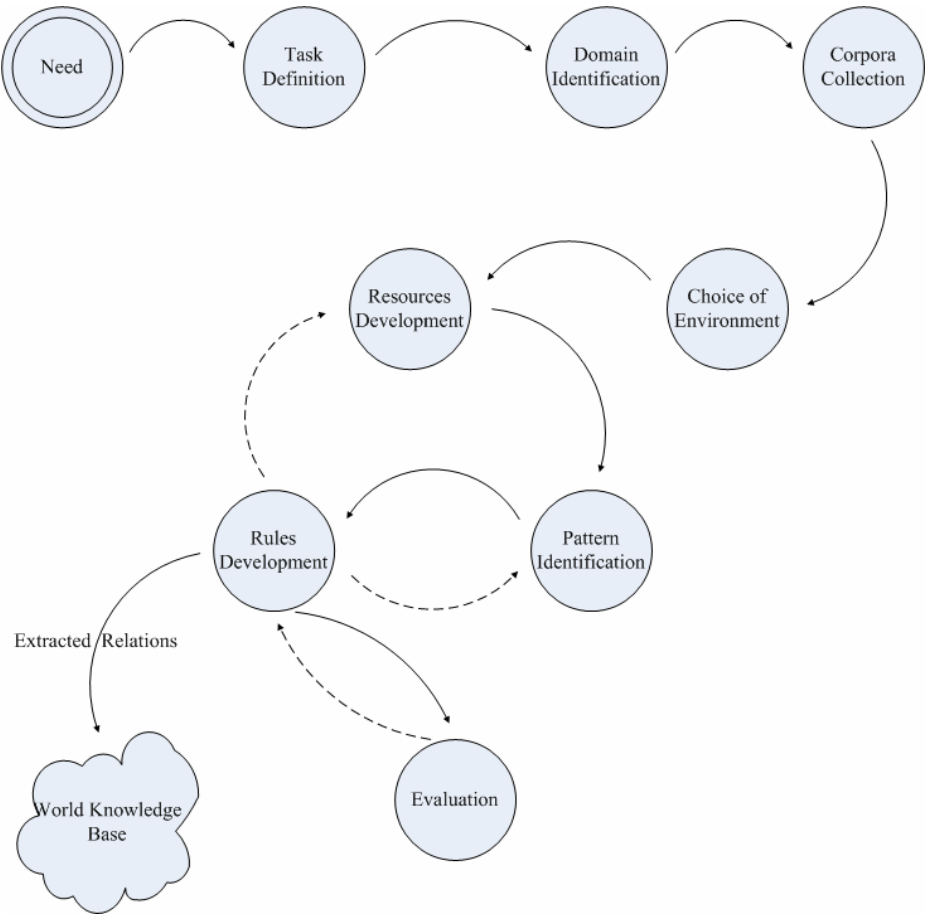


Fig. 1. Methodology flow chart for development of a relation extraction module

entities is that in some domains these relationships are often expressed in various ways in different sources and this allows the RE development to focus only on some of their mentions and be sure that at least one of its mentions will be recognised. This gives a certain quantity of the recognition avoiding processing of complex and ambiguous expressions. The applications in the Semantic Web use some kind of knowledge representation (e.g. ontologies) to associate the extraction results with their model of the world, which is optionally enriched with automatic processing. This instance data might be stored in a semantic repository. In this case it would be enough to recognise even one mention of a relation and store it in the repository enriching the model, rather than attempting to find out all mentions in the texts.

Our particular application aims to recognise three types of relations:

- {Organisation activeIn Location} - organisation related to a particular place
- {Person has Position within Organisation} - a person and an organisation are related to each other through a new entity: Position

- {Person has Position within Location} is an interesting deviation from the person-position-organization pattern and depicts the relation between a person and a location authority of some kind (e.g. King Wangchuck of Bhutan).

3.2 Domain Identification

After identifying the task one can focus on the domain, with which he is going to work. The target domain for the relation extraction in the example application is chosen to be international news. The news articles are an interesting example of being an open domain that mentions entities of all spheres of life, but still having a pretty limited ways of expressing certain relations. This gives us the opportunity to experiment with a real application of the proposed methodology.

3.3 Development and Training Corpora

The collection and pre-processing of the corpora for the development and evaluation phases of any IE task have significant impact on the final result. Two corpora are used in this methodology – one for development and one for evaluation. The corpora need to be annotated with the expected resulting annotations by a human, who can optionally provide corrections to the pre-requisite annotations (output of pre-processing) usually generated by an automatic process.

For the purpose of developing the RE module for the example application 400 BBC News articles have been chosen. Another corpus consisting of 220 documents with CNN, Associated Press, BBC, and Financial Times news has been collected for the final evaluation. Before the human annotation with the three types of relations, the ANNIE system has been used to suggest the named entities. About a person week was necessary for the human-annotation of the corpora.

3.4 Choice of Environment

The environment for a RE module is formed by the surrounding technologies that are used for pre- and post-processing of the result. The current methodology aims at giving general guidelines on what are the steps to successful RE development, but is heavily grounded in the GATE system environment and also the KIM Platform to some extent.

3.5 Resources Development

After having identified the task, domain and relations in focus, it is important to find out the lexical resources that would be helpful in the recognition process. Depending on the specific application needs one can create them from scratch or can collect already existing sources and reuse the efforts of others.

In the example application there are many dependencies on the resources that have been previously gathered. On the one hand, these are the gazetteer lists of the ANNIE system that cover a lot of names of real-world entities and also lexical resources that aid the NE recognition. Another resource in use is the PROTON³ base-upper level

³ <http://proton.semanticweb.org>

ontology that provides a model of about 250 Entity classes of general importance and relations between them. The final model of the extracted relations is produced with respect to this ontology. The Lexical Resource part of the ontology is also used to express the prerequisites for the analysis of the patterns (e.g. keywords that identify *professions* or *job positions*). One more resource used is the World Knowledge Base, a part of the KIM Platform. The WKB contains more than 40 000 entities of general importance and the relations between those, associated with the corresponding ontology classes. These entities are instances of classes like companies, people, organisations, brands, locations, etc.

3.6 Extracting the Relations

One can decide on various approaches for RE module development. The choice of developing a pattern-matching grammars application has been done on the basis of existing expertise within the team and the limited period for the bootstrapping. The GATE platform provides the option to define pattern-matching rules with optional Java code in them and a couple of transducers of these rules with different performance levels. Other options are also possible and are not a deviation from the methodology (for example statistical methods). In section 4 a detailed explanation of the techniques and tools used in the example application is given.

3.7 Evaluation Cycle

In the process of extracting the relations some feedback is useful to be given to the developer. A first test evaluation should be done in the middle of the work process. Analysing the results (seeing the missing annotations) is useful for revision and improvement so that better results are achieved in the final evaluation.

4 RE Approach in the Example Application

In the example application a pattern-matching rules (or grammars) are developed for RE. This section provides an overview of the different techniques used in this process.

4.1 Pattern Identification and Rule Development

Before developing pattern-matching rules one should analyse the training corpus with the idea of finding out the patterns that describe the relations before the actual implementation of a RE module. This could be done the hard way by browsing the text and trying to figure out what the patterns are, or advanced entity indexing technologies could be used. Since the patterns that represent a relation are usually formed by entity references and optionally some keywords, it is of great help for the development if one uses a tool that indexes the textual content with respect to the entity references and allows for the definition of pattern queries.

In the chosen environment, fortunately, there is such a system called ANNIC. ANNIC is based-on-GATE indexing and retrieval search engine, indexing documents by content, linguistic annotations, and features [1]. In the beginning simple queries consisting of the entity types of interest can be put in a broad unspecific way (e.g.

“Organization (any token) (any token) (any token) Location”). The results gave a first impression of the different syntactic combinations between the entities, which was very helpful for the very initial development of the rules.

The rules are written using JAPE - a Java Annotation Patterns Engine. The JAPE grammars comprise phases, which consist of different rules. Each rule consists of a left (specifying the matching pattern)- and right (manipulating the matched pattern)-hand side. The Left Hand Side (LHS) matches the already recognised entities during the NER phase and forms the bound pattern (example of a LHS is given in *Example 1* below). The whole pattern forming the left hand side of the rules can be queried in ANNIC. The action in the Right Hand Side (RHS) of the rules takes the entity identifiers of each of the matched in the LHS entities, and uses them as features of the relation annotation, which is to be generated. By testing the grammars with GATE, one could also see how the different rules affect each other. More information about writing JAPE rules could be found in [7].

Example 1:

```
// Netherlands-based ABN Amro
(({Location}):location
 ({Token.string == "-"})? ({Token.string == "based"})?
 ({Organization}):org
): locorg
```

The rest of this section presents a part of the rule development process and mainly the development of the LHS of the patterns for the example application.

4.1.1 The Use of Unspecified Tokens and Categories

If one wants to include unspecified tokens or any lowercase letters into the pattern instead of well specified ones (e.g. different strings such as “of”, “-”, and “based”), then more testing and analysis of the text is needed, in order to prevent the rule from matching wrong pieces of text. For instance, in the example application in order to leave three lowercase tokens in the LHS of one of the rules it was tested several times with ANNIC with three and after that with four lowercase tokens, and only correct relations were obtained. It was concluded that the construction allowed such usage of unspecified tokens - other syntactic constructions returned wrong relations even with one unspecified token between the entity types.

The ideas are almost the same when using part of speech (POS) categories - whether to use a category (considering the whole particular set of words), or a single string. Usually, it is useful to use more general tokens, because the grammars match more patterns and the recall goes up, but this could be done very carefully, because there can be a negative impact on the precision - some justifiable risk could be taken after the grammars are tested well enough.

4.1.2 Use of Context

Since the annotations, which are going to be generated always cover the span of text from the beginning of the first entity to the end of the last entity in the relation (i.e. **Location’s ORG**), usually the pattern on the left starts and ends also with an entity (see *Example 1* above). However, not always the LHS of the rule has to start or end with a type of entity. Rather, it is sometimes useful to include some context (some

text) before or after the entities as an obligatory part of the LHS. This leads to restriction of the amount of patterns to be matched. For instance, in *Example 2* the part of text, which is of interest, is from the beginning of the *Organization* entity to the end of the *Location* entity, but the pattern obligatory matches also some string (or strings) before the *Organization*'s name.

Example 2:

```
// ... that BMW, one of Germany's most profitable ...
(( ({Token.orth=="lowercase"})?
  | {Token.category==IN}           // preposition
  | {Token.string=="."}
  | ({Token.string=="as"} {Token.string=="such"})
  | {Token.string=="including"}
)
({Organization}): org
({Token.string==","} ({Token.category=="DT"})?
({Token.string=="one"} ( {Token.string=="of"} )? )?
({Token.string=="the"})? ({Location}):location
): locorg
```

4.1.3 Overlapping Relations

Other frequently occurring relations are those in which one entity is related to two or more other entities of the same type – e.g. *Location*'s **ORG** and **ORG**. In this case more than one annotation could be generated in the RHS of the rules.

Various similar combinations could be covered; for example, when two different persons are related to their positions in one organization (see *Example 3*).

Example 3:

```
// BBC's correspondent Greg Wood, and director General
Greg Dyke
({Organization}
({Token.string=="'s"})?
({Token})? ({JobTitle}):job1
({Token.string==","})?
({Token})? ({Person}):person1
(
  ({Token.string==","})? ({Token.string=="and"})?
  ({Token})? ({JobTitle}):job2
  ({Token.string==","})?
  ({Token})? ({Person}):person2
)?): orgperson
```

4.1.4 The Use of Key Words

Other syntactic constructions are too complex to be matched, because of the huge number of strings (tokens) between the types of entities, which makes it more difficult to conclude that the entities are in correct relation. We found it useful to include a particular key word (e.g. “company”, “firm”, “bank”) which is to be one of the

obligatory matching words in the patterns. Thus, a bigger number of unspecified words would be included.

Organization,(determiner)? (Token)?(Token)? (**company**) (Token)? (in) (Location)
 // *Dubai Ports World*, a state-owned **company** based in the *United Arab Emirates*

Much text analysis and testing, however, is needed in order to decide on the key words. This method could also be combined with anaphora resolution techniques (see section 6).

4.1.5 Linking Semantically Related Words – The Location Name with Its Location Adjective

During the work process, it was noticed that there are many constructions for the relation of type *{Person has Position within Location}* using country adjective instead of country name (Spanish instead of Spain). However, if we had “*the Dutch queen Beatrix*” the system could not make a conclusion that *Beatrix* has something to do with *the Netherlands*, because the country name and the corresponding adjective were not formally linked. A way to make a connection is to add a new property in the ontology to the country class, let say “*the Netherlands {has Country Adjective} Dutch*”.

The results of testing two different sets of rules (one with and the other without considering matching of country adjective) showed that it is meaningful to link the country name and the corresponding adjective. Both sets yielded almost the same score for precision; however, the recall percentage for the grammars considering adjectives is 50 % in comparison to 34 % for the other grammars. This implied the presence of many syntactic constructions using the country adjective.

It is possible to modify the rules so that other types of relations between these entities are extracted or even relations between other entities. In these cases the corpora have to be examined for the different corresponding syntactic constructions.

5 Evaluation Results of the Example Application

In this section an overview of the results obtained after the evaluation of the example application is provided. For this application the time invested for the development of the pattern-matching rules was about one person month. In the middle of the work process after some rules for the different types of relations were developed, a first test evaluation over the human-annotated training corpus was made. Analysing the results (seeing the missing annotations) was useful for the revision and improvement of the grammars. Then, the cycle of checking the rules with ANNIC and GATE was repeated several times before making the final evaluation with another human-annotated corpus. Evaluation was performed with the Corpus Benchmark Tool of Gate.

The evaluation measures of the example application are presented in *Table 1*.

{Organization activeIn Location} = activeIn
 {Person has Position within Organization} = hasPosition
 {Person has Position within Location} = hasPosWLoc

Table 1. Final Evaluation

Annotation Type	Correct	Partially Correct	Missing	Spurious	Pr.	R.	F-m.
<i>activeIn</i>	41	3	35	8	0.82	0.54	0.65
<i>hasPosition</i>	80	4	75	2	0.95	0.52	0.67
<i>hasPosWLoc</i>	25	8	25	6	0.74	0.5	0.60

Overall average precision: 0.90

Overall average recall: 0.53

Overall average F-Measure: 0.56

The precision shows how good the grammar rules are in what they are created for – how “precise” the patterns match. From the recall it could be concluded how many of the possible syntactic constructions the grammars are able to cover. The achievement of high precision indicates that if the system matches a pattern, then the probability that the relation is correct is high as well. Moreover, as already mentioned in section 3.1, a certain relation could be represented with different syntactic constructions, therefore, it is more important to be able to extract the relations correctly (high precision), than to extract all the different patterns for a certain relation (high recall).

It is important to note that in order to eliminated the effect of the automatic NER and evaluate properly the effectiveness only of the newly created grammar rules for relation extraction, the human annotated corpus was used for evaluation of the JAPE grammars, deleting only the relation annotations (but leaving the named entity annotations), which the human had made. A JAPE transducer of the grammars was run over this corpus.

6 Future Work

Several ideas which could eventually be useful for the further improvement of the pattern-matching rules are mentioned in this section. Some of them appeared during the work process, but because of different reasons it was considered not to be implemented right away. Others came out after a look through the results returned after the final testing (evaluation) of the grammars.

One improvement would be to make the system recognize a job position from a verb, which will be useful for finding more relations linking, for example, person, organization, and person’s job position in that organization. Here are some examples:

ORG is founded by PERSON → *JOB POSITION = founder*

ORG is headed by PERSON → *JOB POSITION = head*

PERSON leads ORG → *JOB POSITION = leader*

If the job position is recognized in these examples, then an annotation of type *{Person has Position within Organization}* would be generated.

Second, in the knowledge base the different positions are given only in singular forms: i.e. director, Prime Minister, leader, etc. If the system detects the plural forms

of the job positions then more overlapping relations (4.1.3) could be caught. For example, in: “*CAO's head of finance, Tiong Sun, and directors Jia Changbin, Li Yongji and Gu Yanfei*” it will be concluded that the organization (CAO) has three directors: *Jia Changbin is a director of CAO; Li Yongji is a director of CAO; Gu Yanfei is a director of CAO.*

Third, some anaphora resolution methods could be used for catching relations, whose entities are scattered over a wider span of text. Further analysis of the text would be made, so that if the entities, which are to be related, are not close enough and if another word referring to the entity is used with the other entities, then the system can recognize that the two words (the entity in question and its anaphor) represent the same thing. For example, in “*Google reiterated its policy against making earnings projections. Company co-founders Larry Page and Sergey Brin ...*” a relation between Google and its two co-founders (relation of type {*Person has Position within Organization*}) could be made. For this purpose **Company** and its antecedent **Google** should be recognized as the same thing.

Regarding the methodology, we would like to enrich it with any new phase or approach that proves to be beneficial. One particular thing, which can be considered in the evaluation cycle, is the use of a training corpus consisting of documents from various resources (such as those in the testing corpus). Thus, the probability to achieve better results for the recall would be higher. Further, one of our ideas is to generalize the methodology by covering also machine learning and other techniques for relation extraction, and not only rule-based ones.

7 Related Work

There are different approaches to identify a relationship between named entities in texts. In [8] a machine learning approach to relation extraction is presented, and a methodology based on kernel methods. Relations between people, organizations, and locations are looked for. Considerable results for recall and precision are achieved. Other interesting machine learning approaches also using kernel functions are proposed in [9] and [3]. In [9] the focus is on recognizing gene/protein interactions from biomedical literature. Shallow linguistic processing is considered, and the RE is treated as a classification problem. In [3] an attempt of trying to make use of the different information provided from each of the processing steps in the system is made.

In [2] a development of a framework for extracting entities and relations is proposed. Also considering Information Extraction techniques and the use of some structural knowledge representation schema (ontologies), an idea of using automated methods for the enrichment of documents with a semantic layer is adopted, so that the documents become machine-readable and easily processed by different Semantic Web applications. In [5] adaptive information extraction considering the use of sublanguages for particular domains is presented, and certain methods, making emphasis on the importance of linguistic analysis techniques and tools for extracting specific information are described.

In [4] and [6] a nice overview of different methods and approaches of how to extract specific information is given. Typical problems in the process of information extraction are illustrated, as well as a motivation for the development of new algorithms in this area.

8 Conclusion

The bootstrapping of a relation extraction module for the three exemplified relation types proved to be very successful and this success is due to the methodology that has been presented here. Following the defined steps within the chosen environment ensures the fast and robust development. The time invested was about one person month which also shows the good choice of environment and the qualities of the presented methodology.

From the perspective of the RE module, a balance between the desire to identify as many relations as possible (e.g. by generalizing the rules in the case of the example application), and the risk of catching wrong relations, should be looked for. Many new ideas come out from spotting the common types of the returned missing annotations after the evaluation. In this sense, repeating the cycle of evaluating the chosen approach and testing with human-annotated corpus (or a very well automatically annotated one) would give better results.

References

1. Aswani N, Tablan V, Bontcheva K, Cunningham H, Indexing and Querying Linguistic Metadata and Document Content. In *RANLP 2005*, 21-23 September 2005, Borovets, Bulgaria
2. Iria J., Ciravegna F, Relation Extraction for Mining the Semantic Web. In *Proceedings Machine Learning for the Semantic Web Dagstuhl*, Seminar 05071, Dagstuhl, DE
3. Zhao S, Grishman R, Extracting Relations with Integrated Information Using Kernel Methods, *ACL 2005*
4. Agichtein E, Scaling Information Extraction to Large Document Collections,” In *the IEEE Data Engineering Bulletin Special Issue on “Searching and Mining Digital Libraries”*, Dec. 2005
5. Grishman R, Adaptive Information Extraction and Sublanguage Analysis, In *Proceedings of the Workshop on Adaptive Text Extraction and Mining at the 17 International Joint Conference on Artificial Intelligence*, 2001
6. Grishman R Information extraction: Techniques and challenges. In *Proceedings of the Information Extraction International Summer School SCIE-97*, M. T. Pazienza, ed. New York: Springer-Verlag
7. GATE User guide, <http://gate.ac.uk/sale/tao/index.html>
8. Zelenko D, Aone C, Richardella A, Kernel Methods for Relation Extraction, *J. Mach. Learn. Res.*, Vol. 3 (2003), pp. 1083-1106.
9. Claudio G, Lavelli A, Romano L, Exploiting Shallow Linguistic Information for Relation Extraction from Biomedical Literature, In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, Trento, Italy, 3-7 April 2006

Using Verbs to Characterize Noun-Noun Relations

Preslav Nakov and Marti Hearst

EECS and School of Information
University of California at Berkeley
Berkeley CA 94720, USA
{nakov@cs, hearst@sims}.berkeley.edu

Abstract. We present a novel, simple, unsupervised method for characterizing the semantic relations that hold between nouns in noun-noun compounds. The main idea is to discover *predicates* that make explicit the hidden relations between the nouns. This is accomplished by writing Web search engine queries that restate the noun compound as a relative clause containing a wildcard character to be filled in with a verb. A comparison to results from the literature suggest this is a promising approach.

Keywords: Web statistics, noun compound, lexical semantics, componential analysis.

1 Introduction

An important characteristic of technical literature is the abundance of long noun compounds like *bone marrow biopsy specimen* and *neck vein thrombosis*. While eventually mastered by domain experts, their interpretation poses a significant challenge for automated analysis, e.g., what is the relationship between *bone marrow* and *biopsy*? Between *biopsy* and *specimen*? Understanding relations between multiword expressions is important for many tasks, including question answering, textual entailment, machine translation, and information retrieval, among others.

In this paper we focus on the problem of determining the semantic relation(s) that holds within two-word English noun compounds. We introduce a novel approach for this problem: use paraphrases posed against an enormous text collection as a way to determine which predicates, represented as verbs, best characterize the relationship between the nouns.

Most algorithms that perform semantic interpretation place heavy reliance on the appearance of verbs, since they are the predicates which act as the backbone of the assertion being made. Noun compounds are terse elisions of the predicate; their structure assumes that the reader knows enough about the constituent nouns and about the world at large to be able to infer what the relationship between the words is. Our idea is to try to uncover the relationship between the noun pairs by, in essence, rewriting or paraphrasing the noun compound in such a

way as to be able to determine the predicate(s) holding between the nouns. What is especially novel about this approach is paraphrasing noun compound semantics in terms of concrete verbs, rather than a fixed number of abstract predicates (e.g., HAVE, MAKE, USE), relations (e.g., LOCATION, INSTRUMENT, AGENT), or prepositions (e.g., OF, FOR, IN), as is traditional in the literature.

This idea builds on earlier work which shows that the vast size of the text available on the Web makes it likely that the same concept is stated in many different ways [1,2]. This information in turn can be used to solve syntactic ambiguity problems [3,4]. Here we extend that idea by applying it to determining semantic relations.

In our approach, we pose paraphrases for a given noun compound by rewriting it as a phrase that contains a wildcard where the verb would go. For example, we rewrite *neck vein* as "**vein that * neck**", send this as a query to a Web search engine, and then parse the resulting snippets to find the verbs that appear in the place of the wildcard. Some of the most frequent verbs (+ prepositions) found for *neck vein* are: *emerge from*, *pass through*, *be found in*, *be terminated at*, *be in*, *flow in*, *run from*, *terminate in*, *descend in*, *come from*, etc. A comparison to examples from the literature suggest this is a promising approach with a broad range of potential applications.

In the remainder of this paper we first describe related work, then give details of the algorithm, present preliminary results as compared to other work in the literature, and discuss potential applications.

2 Related Work

There is currently no consensus as to which set of relations should hold between nouns in a noun compound, but most existing approaches make use of a set of a small number of abstract relations, typically less than 50. However, some researchers (e.g., Downing [5]), have proposed that an unlimited number is needed; in this paper we will hold a similar position.

One of the most important theoretical linguistic models is that of Levi [6], which states that noun compounds are derived through two basic processes: predicate nominalization (e.g., '*The president refused the appointment.*' → *presidential refusal*) and predicate deletion ('*pie made of apples*' → *apple pie*). According to Levi, predicate nominalizations can be subjective or objective, depending on whether the subject or the object is retained, and the relation can be further classified as ACT, PRODUCT, AGENT or PATIENT, depending on the thematic role between the nominalized verb and the retained argument. Predicate deletion, in turn, is limited to the following abstract predicates: five verbs (CAUSE, HAVE, MAKE, USE and BE) and four prepositions (IN, FOR, FROM and ABOUT). For example, according to Levi, *night flight* should be analyzed as IN (*flight at night*), and *tear gas* as CAUSE (*gas that causes tears*). A problem with this approach is that the predicates are too abstract, and can be ambiguous, e.g., *sand dune* is both HAVE and BE.

Lauer [7] simplifies Levi's idea by redefining the semantic relation identification problem as one of predicting which among the 8 prepositions is most likely to

be associated with the compound when rewritten: *of*, *for*, *in*, *at*, *on*, *from*, *with* and *about*. Lapata and Keller [1] improve on Lauer's results (whose accuracy was 40%) by using his problem definition along with Web statistics to estimate (*noun*₁, *prep*, *noun*₂) trigram frequencies, achieving 55.71% accuracy. However, the preposition-oriented approach is problematic because the same preposition can indicate several different relations, and conversely, the same relation can be indicated by several different prepositions. For example, *in*, *on*, and *at* can all refer to both LOCATION and TIME.

Rosario and Hearst [8] show that a discriminative classifier can work quite well at assigning relations from a pre-defined set if training data is supplied in a domain-specific setting (60% accuracy, 18 classes). In later work, [9] provide a semi-supervised approach for characterizing the relation between two nouns in a bioscience noun-noun compound based on the semantic category each of the constituent nouns belongs to. Although this "descent of hierarchy" approach achieved a precision of 90% for finding the correct level of generalization, it does not assign *names* to the relations.

Girju et al. [10] apply both classic (SVM and decision trees) and novel supervised models (semantic scattering and iterative semantic specialization), using WordNet, word sense disambiguation, and a set of linguistic features. They test their system against both Lauer's 8 prepositional paraphrases and another set of 21 semantic relations, achieving up to 54% accuracy on the latter.

Lapata [11] focuses on the disambiguation of nominalizations. Using partial parsing, sophisticated smoothing and contextual information, she achieved 86.1% accuracy (baseline 61.5%) on the binary decision task of whether the modifier used to be the subject or the object of the nominalized verb (the head).

Girju et al. [12] present an SVM-based approach for the automatic classification of semantic relations in nominalized noun phrases (where either the head or the modifier has been derived from a verb). Their classification schema consists of 35 abstract semantic relations and has been also used by [13] for the semantic classification of noun phrases in general.

Turney and Littman [14] characterize the relation between two words, *X* and *Y*, as a vector whose coordinates correspond to Web frequencies for 128 phrases like "*X for Y*", "*Y for X*", etc., derived from a fixed set of 64 joining terms (e.g. "*for*", "*such as*", "*not the*", "*is **", etc.). These vectors are then used in a nearest-neighbor classifier, which maps them to a set of fixed relations. He achieved an F-value of 26.5% (random guessing 3.3%) with 30 relations, and 43.2% (random: 20%) with 5 relations.

In work to appear, Turney [15] presents an unsupervised algorithm for mining the Web for patterns expressing implicit semantic relations. For example, CAUSE (e.g. *cold virus*) is best characterized by "*Y * causes X*", and "*Y in * early X*" is the best pattern for TEMPORAL (e.g. *morning frost*). He obtains an F-value 50.2% for 5 classes. This approach is the closest to our proposal.

Most other approaches to noun compound interpretation used hand-coded rules for at least one component of the algorithm [16], or rules combined with lexical resources [17] (52% accuracy, 13 relations). [18] make use of the identity

of the two nouns and a number of syntactic clues in a nearest-neighbor classifier with 60-70% accuracy.

3 Using Verbs to Characterize Noun-Noun Relations

As we have described above, traditionally, the semantics of noun compounds have been represented as a set of abstract relations. This is problematic for several reasons. First, it is unclear which is the best set, and mapping between different sets has proven challenging [10]. Second, being both abstract and limited, these sets only capture a small part of the semantics, often multiple meanings are possible, and sometimes none of the pre-defined meanings are suitable for a given example. Finally, it is unclear how useful the proposed sets are, as researchers have often fallen short of demonstrating practical uses.

We believe verbs have more expressive power and are better tailored for the task of semantic representation: there is an infinite number of them (according to [5]) and they can capture fine-grained aspects of the meaning. For example, while *wrinkle treatment* and *migraine treatment* express the same abstract relation TREATMENT-FOR-DISEASE, some fine-grained differences can be shown by specific verbs e.g., *smooth* is possible in a verbal paraphrase of the former, but not of the latter.

In many theories, verbs play an important role in the process of noun compound derivation, and they are frequently used to make the hidden relation overt. This allows not only for simple and effective extraction (as we have seen above), but also for straightforward uses of the extracted verbs and paraphrases in NLP tasks like machine translation, information retrieval, etc.

We further believe that a single verb often is not enough and that the meaning is approximated better by a collection of verbs. For example, while *malaria mosquito* can very well be characterized as CAUSE (or *cause*), further aspects of the meaning, can be captured by adding some additional verbs e.g., *carry*, *spread*, *transmit*, *be responsible for*, *be infected with*, *pass on*, etc.

In the next section, we describe our algorithm for discovering predicate relations that hold between nouns in a compound.

4 Method

In a typical noun-noun compound “*noun₁ noun₂*”, *noun₂* is the head and *noun₁* is a modifier, attributing a property to it. Our idea is to preserve the head-modifier relation by substituting the pre-modifier *noun₁* with a suitable post-modifying relative phrase; e.g., “*tear gas*” can be transformed into “*gas that causes tears*”, “*gas that brings tears*”, “*gas which produces tears*”, etc. Using all possible inflections of *noun₁* and *noun₂* as found in WordNet [19], we issue exact phrase Google queries of the following type:

"noun2 THAT * noun1"

where THAT can be *that*, *which* or *who*. The Google * operator is a one-word wildcard substitution; we issue queries with up to 8 stars.

We collect the text snippets (summaries) from the search results pages (up to 1000 per query) and we only keep the ones for which the sequence of words following **noun1** is non-empty and contains at least one non-noun, thus ensuring the snippet includes the entire noun phrase. To help POS tagging and shallow parsing of the snippet, we further substitute the part before **noun2** by the fixed phrase “*We look at the*”. We then perform POS tagging [20] and shallow parsing¹, and extract the verb (and the following preposition, if any) between **THAT** and **noun1**. We allow for adjectives and participles to fall between the verb and the preposition, but not nouns; we ignore the modals, and the auxiliaries, but retain the passive *be*, and we make sure there is exactly one verb phrase (thus disallowing complex paraphrases like “*gas that makes the eyes fill with tears*”). Finally, we convert the main verb to an infinitive using WordNet [19].

The proposed method is similar to previous paraphrase acquisition approaches which search for similar/fixed endpoints and collect the intervening material. Lin and Pantel [21] extract paraphrases from dependency tree paths whose ends contain similar sets of words by generalizing over these ends. For example, for “*X solves Y*” they extract paraphrasing templates like “*Y is resolved by X*”, “*X resolves Y*”, “*X finds a solution to Y*” and “*X tries to solve Y*”. The idea is extended by Shinyama et al. [22], who use named entities of matching semantic class as anchors, e.g., **LOCATION**, **ORGANIZATION**, etc. However, the goal of these approaches is to create summarizing paraphrases, while we are interested in finding noun compound semantics.

Table 1 shows a subset of the verbs found using our extraction method for *cancer treatment*, *migraine treatment*, *wrinkle treatment* and *herb treatment*. We can see that *herb treatment* is very different from the other compounds and shares no features with them: it *uses* and *contains* herb, but does not *treat* it. Further, while migraine and wrinkles cannot be *cured*, they can be *reduced*. Migraines can also be *prevented*, and wrinkles can be *smoothed*. Of course, these results are merely suggestive and should not be taken as ground truth, especially the absence of indicators. Still they seem to capture interesting fine-grained semantic distinctions, which normally require deep knowledge of the semantics of the two nouns and/or about the world.

5 Evaluation

5.1 Comparison with Girju et al., 2005

In order to test this approach, we compared it against examples from the literature. In this preliminary evaluation, we manually determined if verbs accurately reflected each paper’s set of semantic relations.

Table 3 shows the results comparing against the examples of 21 relations that appear in [10]. In two cases, the most frequent verb is the copula, but the following most frequent verbs are appropriate semantic characterizations of the compound. In the case of “*malaria mosquito*”, one can argue that the **CAUSE**

¹ OpenNLP tools: <http://opennlp.sourceforge.net>

Table 1. Some verbs found for different kinds of treatments

	cancer treatment	migraine treatment	wrinkle treatment	herb treatment
<i>treat</i>	+	+	+	—
<i>prevent</i>	+	+	—	—
<i>cure</i>	+	—	—	—
<i>reduce</i>	—	+	+	—
<i>smooth</i>	—	—	+	—
<i>cause</i>	+	—	—	—
<i>contain</i>	—	—	—	+
<i>use</i>	—	—	—	+

Table 2. Example componential analysis for *man*, *woman*, *boy* and *bull*

	man	woman	boy	bull
ANIMATE	+	+	+	+
HUMAN	+	+	+	—
MALE	+	—	+	+
ADULT	+	+	—	+

relation, assigned by [10] is not really correct, in that the disease is only indirectly caused by the mosquitos, but rather is carried by them, and the proposed most frequent verbs *carry* and *spread* more accurately represent an AGENT relation. Nevertheless, *cause* is the third most frequent verb, indicating that it is common to consider the indirect relation as causal. In the case of *combustion gas*, the most frequent verb *support*, while being a good paraphrase of the noun compound, is not directly applicable to the relation assigned by [10] as RESULT, but the other verbs are.

In all other cases shown, the most frequent verbs accurately capture the relation assigned by [10]. In some cases, less frequent verbs indicate other logical entailments from the noun combination.

For the following examples, no meaningful verbs were found (in most cases there appears not to be a meaningful predicate for the particular nouns paired, or a nominalization plays the role of the predicate): *quality sound*, *crew investigation*, *image team*, *girl mouth*, *style performance*, *worker fatalities*, and *session day*.

5.2 Comparison with Barker and Szpakowicz, 1998

Table 4 shows comparison to examples from [18]. Due to space limitations, here we discuss the first 8 relations only. We also omitted *charitable donation* and *overdue fine*, as the modifier in these cases is an adjective, and *composer arranger*, because no results were found.

We obtain very good results for AGENT and INSTRUMENT, but other relations are problematic, probably because the assigned classifications are of varying

Table 3. Comparison to examples (14 out of 21) found in [10], showing the most frequently extracted verbs. Verbs expressing the target relation are in bold, those referring to a different, but semantically valid, are in italic, and errors are struck out.

Sem. relation	Example	Extracted Verbs
POSSESSION	<i>family estate</i>	be in (29), be held by (9), be owned by (7)
TEMPORAL	<i>night flight</i>	arrive at (19), leave at (16), be at (6), be conducted at (6), occur at (5)
IS-A (HYPERNYMY)	<i>Dallas city</i>	include (9)
CAUSE	<i>malaria mosquito</i>	<i>carry</i> (23), <i>spread</i> (16), cause (12), <i>transmit</i> (9), <i>bring</i> (7), <i>have</i> (4), <i>be infected with</i> (3), be responsible for (3), <i>test positive for</i> (3), infect many with (3), <i>be needed for</i> (3), pass on (2), give (2), give out (2)
MAKE/PRODUCE	<i>shoe factory</i>	produce (28), make (13), manufacture (11)
INSTRUMENT	<i>pump drainage</i>	be controlled through (3), use (2)
LOCATION/SPACE	<i>Texas university</i>	be (5), be in (4)
PURPOSE	<i>migraine drug</i>	treat (11), be used for (9), prevent (7), work for (6), stop (4), help (4), work (4), be prescribed for (3), relieve (3), block (3), <i>be effective for</i> (3), <i>be for</i> (3), help ward off (3), <i>seem effective against</i> (3), end (3), reduce (2), cure (2)
SOURCE	<i>olive oil</i>	come from (13), be obtained from (11), be extracted from (10), be made from (9), be produced from (7), be released from (4), <i>taste like</i> (4), be beaten from (3), be produced with (3), emerge from (3)
TOPIC	<i>art museum</i>	focus on (29), <i>display</i> (16), bring (14), highlight (11), <i>house</i> (10), <i>exhibit</i> (9), demonstrate (8), feature (7), <i>show</i> (5), tell about (4), cover (4), concentrate in (4)
MEANS	<i>bus service</i>	use (14), operate (6), <i>include</i> (6)
EXPERIENCER	<i>disease victim</i>	spread (12), acquire (12), suffer from (8), die of (7), <i>develop</i> (7), contract (6), catch (6), be diagnosed with (6), <i>have</i> (5), <i>beat</i> (5), be infected by (4), survive (4), die from (4), get (4), pass (3), fall by (3), <i>transmit</i> (3), <i>avoid</i> (3)
THEME	<i>car salesman</i>	sell (38), mean inside (13), buy (7), travel by (5), pay for (4), deliver (3), push (3), demonstrate (3), purr (3), bring used (3), <i>know more about</i> (3), <i>pour through</i> (3)
RESULT	<i>combustion gas</i>	<i>support</i> (22), result from (14), be produced during (11), be produced by (8), be formed from (8), form during (8), be created during (7), originate from (6), be generated by (6), develop with (6), come from (5), be cooled (5)

Table 4. Comparison to examples (8 out of 20) from [18], showing the most frequently extracted verbs. Verbs expressing the target relation are in bold, those referring to a different, but semantically valid, are in italic, and errors are struck out.

Relation	Example	Extracted Verbs
AGENT	<i>student protest</i>	be led by(6) , be sponsored by(6) , pit(4) , <i>be(4)</i> , be organized by(3) , be staged by(3) , be launched by(3) , be started by(3) , be supported by(3) , <i>involve(3)</i> , <i>arise from(3)</i>
AGENT	<i>band concert</i>	<i>feature(17)</i> , <i>capture(10)</i> , <i>include(6)</i> , be given by(6) , <i>play of(4)</i> , <i>involve(4)</i> , be than(4) , be organized by(3) , be by(3) , <i>start with(3)</i> , <i>bring(3)</i> , <i>take(3)</i> , <i>consist of(3)</i>
AGENT	<i>military assault</i>	be initiated by(4) , <i>shatter(2)</i>
BENEFICIARY	<i>student price</i>	<i>be(14)</i> , mean(4) , differ from(4) , be unfair for(3) , be discounted for(3) , be for(3) , be affordable for(3) , be charged for(3) , please(3) , be shared with(3) , <i>draw in(3)</i>
CAUSE	<i>exam anxiety</i>	<i>be generated during(3)</i>
CONTAINER	<i>printer tray</i>	hold(12) , <i>come with(9)</i> , <i>be folded(8)</i> , <i>fit under(6)</i> , be folded into(4) , pull from(4) , be inserted into(4) , <i>be mounted on(4)</i> , <i>be used by(4)</i> , be inside(3) , feed into(3)
CONTAINER	<i>flood water</i>	<i>cause(24)</i> , <i>produce(9)</i> , <i>remain after(9)</i> , <i>be swept by(6)</i> , <i>create(5)</i> , <i>bring(5)</i> , <i>reinforce(5)</i>
CONTAINER	<i>film music</i>	<i>fit(16)</i> , be in(13) , be used in(11) , be heard in(11) , <i>play throughout(9)</i> , <i>be written for(9)</i>
CONTAINER	<i>story idea</i>	<i>tell(20)</i> , <i>make(19)</i> , <i>drive(15)</i> , <i>become(13)</i> , <i>turn into(12)</i> , <i>underlie(12)</i> , occur within(8) , hold(8) , <i>tie(8)</i> , <i>be(8)</i> , <i>spark(8)</i> , appear throughout(7) , <i>tell(7)</i> , <i>move(7)</i> , <i>come from(6)</i>
CONTENT	<i>paper tray</i>	feed(6) , <i>be lined with(6)</i> , <i>stand up(6)</i> , hold(4) , contain(4) , <i>catch(4)</i> , overflow with(3)
CONTENT	<i>eviction notice</i>	<i>result in(10)</i> , <i>precede(3)</i> , <i>make(2)</i>
DESTINATION	<i>game bus</i>	<i>be in(6)</i> , leave for(3) , <i>be like(3)</i> , <i>be(3)</i> , <i>make playing(3)</i> , <i>lose(3)</i>
DESTINATION	<i>exit route</i>	<i>be indicated by(4)</i> , reach(2) , <i>have(1)</i> , <i>do(1)</i>
DESTINATION	<i>entrance stairs</i>	<i>look like(4)</i> , <i>stand outside(3)</i> , <i>have(3)</i> , <i>follow from(3)</i> , <i>be at(3)</i> , be(3) , <i>descend from(2)</i>
EQUATIVE	<i>player coach</i>	<i>work with(42)</i> , <i>recruit(28)</i> , be(19) , <i>have(16)</i> , <i>know(16)</i> , <i>help(12)</i> , <i>coach(11)</i> , <i>take(11)</i>
INSTRUMENT	<i>electron microscope</i>	use(27) , <i>show(5)</i> , work with(4) , utilize(4) , employ(4) , <i>beam(3)</i>
INSTRUMENT	<i>diesel engine</i>	<i>be(18)</i> , operate on(8) , <i>look like(8)</i> , use(7) , <i>sound like(6)</i> , run on(5) , be on(5)
INSTRUMENT	<i>laser printer</i>	use(20) , <i>consist of(6)</i> , <i>be(5)</i>

quality: *printer tray* and *film music* are probably correctly assigned to **CONTAINER**, but *flood water* and *story idea* are not; *entrance stairs* (**DESTINATION**) could be equally well analyzed as **LOCATED** or **SOURCE**; and *exam anxiety* (**CAUSE**) probably refers to **TIME**. Finally, although we find the verb *be* ranked third for *player coach*, the **EQUATIVES** pose a problem in general, as the copula is not very frequent in this form of paraphrase.

5.3 Comparison with Rosario and Hearst, 2002

As we mentioned above, [9] characterize noun-noun compounds based on the semantic category, in the MeSH lexical hierarchy, each of the constituent nouns belongs to. For example, all noun compounds in which the first noun is classified under the A01 sub-hierarchy² (*Body Regions*), and the second one falls into A07 (*Cardiovascular System*), are hypothesized to express the same relation. Examples include *mesentery artery*, *leg vein*, *finger capillary*, etc.

By contrast, for the category pair A01-M01 (*Body Regions–Persons*) a distinction is needed between different kinds of persons and the algorithm needs to descend one level on the M01 side: M01.643 (*Patients*), M01.898 (*Donors*), M01.150 (*Disabled Persons*).

Table 5 shows some results of our comparison to [9]. Given a category pair (e.g., A01-A07), we consider all of the noun-noun compounds whose elements are in the corresponding MeSH sub-hierarchies, and we acquire a set of paraphrasing verbs+prepositions from the Web for each of them. We then aggregate the results from all such word pairs in order to obtain a set of paraphrasing verbs for the target category pair.

6 Potential Applications

The extracted verbs (+prepositions) have the potential to be useful for a number of important NLP tasks. For example, they may help in the process of noun compound translation [23]. They could be also directly integrated into a paraphrase-augmented machine translation system [24], machine translation evaluation system [25] [26], or summarization evaluation system [27].

Assuming annotated training data, the verbs could be used as features in the prediction of abstract relations like **TIME** and **LOCATION**, as is done by [14] and [15], who used the vector-space model and a nearest-neighbor classifier.

These relations in turn could play an important role in other applications, as demonstrated by [28], who achieved state-of-the-art results on the PASCAL Recognizing Textual Entailment challenge.

In information retrieval, the verbs could be used for index normalization [29] or query refinement, e.g., when querying for *migraine treatment*, pages containing good paraphrasing verbs, like *relieve* or *prevent*, would be preferred.

² In MeSH each concept is assigned one or more codes, corresponding to positions in the hierarchy e.g., A (*Anatomy*), A01 (*Body Regions*), A01.456 (*Head*), A01.456.505 (*Face*), A01.456.505.420 (*Eye*). *Eye* is ambiguous; it is also A09.371 (A09 represents *Sense Organs*).

Table 5. Comparison to [9] showing the most frequent verbs

Categ. Pair	Examples	Extracted Verbs
A01-A07 (Body Regions - Cardiovascular System)	ankle artery foot vein forearm vein finger artery neck vein head vein leg artery thigh vein	feed(133), supply(111), drain(100), be in(44), run(37), appear on(29), be located in(22), be found in(20), run through(19), be behind(19), run from(18), serve(15), be felt with(14), enter(14), pass through(12), pass by(12), show on(11), be visible on(11), run along(11), nourish(10), be seen on(10), occur on(10), occur in(9), emerge from(9), go into(9), ...
A01-M01.643 (Body Regions - Disabled Persons)	arm patient eye outpatient abdomen patient	be(54), lose(40), have(30), be hit in(11), break(9), gouge out(9), injure(8), receive(7), be stabbed in(7), be shot in(7), need(6), ...
A01-M01.150 (Body Regions - Disabled Persons)	leg amputee arm amputee knee amputee	lose(13), grow(6), have cut off(4), miss(2), need(1), receive(1), be born without(1)
A01-M01.898 (Body Regions - Donors)	eye donor skin donor	give(4), provide(3), catch(1)
D02-E05.272 (Organic Chemicals - Diet)	choline diet methionine diet carotene diet saccharin diet	be low in(18), contain(13), be deficient in(11), be high in(7), be rich in(6), be sufficient in(6), include(4), be supplemented with(3), be in(3), be enriched with(3), contribute(2), miss(2), ...

The verbs and prepositions, intervening between the two nouns could be also used to seed a Web search for whole classes of NPs [30], such as diseases, drugs, etc. For example, after finding that *prevent* is a good paraphrase for *migraine treatment*, we can use the query "*** which prevents migraines**" to obtain different treatments/drugs for migraine, e.g. *feverfew*, *Topamax*, *natural treatment*, *magnesium*, *Botox*, *Glucosamine*, etc.

Finally, the extracted verbs could be used for linguistic analysis. Note the similarity between Table 1 and Table 2. The latter shows a sample *componential analysis*, which represents word's semantics in terms of primitives, called components or features, thus making explicit relations like hyponymy, incompatibility, etc. [31,32,33]. Table 1 shows a similar semantic representation for noun-noun compounds. While the classic componential analysis has been criticized for being inherently subjective, a new *dynamic componential analysis* would extract the components automatically from a large corpus in a principled manner.

7 Conclusions and Future Work

We have presented a simple unsupervised approach to noun compound interpretation in terms of predicates characterizing the hidden relation, which could be useful for many NLP tasks.

A significant benefit of our approach is that it does not require knowledge of the meanings of constituent nouns in order to correctly assign relations. A

potential drawback is that it will probably not work well for low-frequency words, so semantic class information will be needed for these cases.

In future we plan to apply full parsing to reduce the errors caused by shallow parsing and POS errors. We will also assess the results against a larger collection of manually labeled relations, and have an independent evaluation of the appropriateness of the verbs for those relations. We also plan to combine this work with the structural ambiguity resolution techniques of [4], and determine semantic relations among multi-word terms. Finally, we want to test the approach on some of the above-mentioned NLP tasks.

References

1. Lapata, M., Keller, F.: Web-based models for natural language processing. *ACM Transactions on Speech and Language Processing* **2** (2005) 1–31
2. Banko, M., Brill, E.: Scaling to very very large corpora for natural language disambiguation. *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics* (2001) 26–33
3. Nakov, P., Hearst, M.: Search engine statistics beyond the n-gram: Application to noun compound bracketing. In: *Proceedings of the 9th Conference on Computational Natural Language Learning*. (2005) 17–24
4. Nakov, P., Hearst, M.: Using the Web as an implicit training set: Application to structural ambiguity resolution. In: *Proceedings of HLT-EMNLP*, Vancouver, British Columbia, Canada (2005) 835–842
5. Downing, P.: On the creation and use of English compound nouns. *Language* **53**(4) (1977) 810–842
6. Levi, J.: *The Syntax and Semantics of Complex Nominals*. Academic Press, New York (1978)
7. Lauer, M.: *Designing Statistical Language Learners: Experiments on Noun Compounds*. PhD thesis, Department of Computing Macquarie University NSW 2109 Australia (1995)
8. Rosario, B., Hearst, M.: Classifying the semantic relations in noun compounds via a domain-specific lexical hierarchy. In Lee, L., Harman, D., eds.: *Proceedings of EMNLP (Empirical Methods in Natural Language Processing)*. (2001) 82–90
9. Rosario, B., Hearst, M., Fillmore, C.: The descent of hierarchy, and selection in relational semantics. In: *Proceedings of ACL*. (2002) 247–254
10. Girju, R., Moldovan, D., Tatu, M., Antohe, D.: On the semantics of noun compounds. *Computer Speech and Language* **19**(4) (2005) 479–496
11. Lapata, M.: The disambiguation of nominalisations. *Computational Linguistics* **28**(3) (2002) 357–388
12. Girju, R., Giuglea, A.M., Olteanu, M., Fortu, O., Bolohan, O., Moldovan, D.: Support vector machines applied to the classification of semantic relations in nominalized noun phrases. In: *Proceedings of the HLT-NAACL Workshop on Computational Lexical Semantics*, Boston, MA (2004) 68–75
13. Moldovan, D., Badulescu, A., Tatu, M., Antohe, D., Girju, R.: Models for the semantic classification of noun phrases. In: *Proceedings of the Computational Lexical Semantics Workshop at HLT-NAACL*. (2004) 60–67
14. Turney, P., Littman, M.: Corpus-based learning of analogies and semantic relations. *Machine Learning Journal* **60**(1-3) (2005) 251–278

15. Turney, P.: Expressing implicit semantic relations without supervision. In: Proceedings of COLING-ACL, Australia (2006)
16. Finin, T.: The Semantic Interpretation of Compound Nominals. Ph.d. dissertation, University of Illinois, Urbana, Illinois (1980)
17. Vanderwende, L.: Algorithm for automatic interpretation of noun sequences. In: Proceedings of COLING-94. (1994) 782–788
18. Barker, K., Szpakowicz, S.: Semi-automatic recognition of noun modifier relationships. In: Proceedings of COLING-ACL'98, Montreal, Canada (1998) 96–102
19. Fellbaum, C., ed.: WordNet: An Electronic Lexical Database. MIT Press (1998)
20. Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of HLT-NAACL 2003. (2003) 252–259
21. Lin, D., Pantel, P.: Discovery of inference rules for question-answering. *Natural Language Engineering* **7**(4) (2001) 343–360
22. Shinyama, Y., Sekine, S., Sudo, K., Grishman, R.: Automatic paraphrase acquisition from news articles. In: Proceedings of Human Language Technology Conference (HLT 2002), San Diego, USA (2002) 40–46
23. Baldwin, T., Tanaka, T.: Translation by machine of complex nominals: Getting it right. In: Proceedings of the ACL04 Workshop on Multiword Expressions: Integrating Processing. (2004) 24–31
24. Callison-Burch, C., Koehn, P., Osborne, M.: Improved statistical machine translation using paraphrases. In: Proceedings of HLT-NAACL 2006. (2006) 17–24
25. Russo-Lassner, G., Lin, J., Resnik, P.: A paraphrase-based approach to machine translation evaluation. Technical Report LAMP-TR-125/CS-TR-4754/UMIACS-TR-2005-57, University of Maryland (2005)
26. Kauchak, D., Barzilay, R.: Paraphrasing for automatic evaluation. In: Proceedings of HLT-NAACL 2006. (2006) 455–462
27. Liang Zhou, Chin-Yew Lin, D.S.M., Hovy, E.: PARAEVAL: Using paraphrases to evaluate summaries automatically. In: Proceedings of HLT-NAACL 2006. (2006) 447–454
28. Tatu, M., Moldovan, D.: A semantic approach to recognizing textual entailment. In: Proceedings of HLT/EMNLP 2005. (2005) 371–378
29. Zhai, C.: Fast statistical parsing of noun phrases for document indexing. In: Proceedings of the fifth conference on Applied natural language processing, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1997) 312–319
30. Etzioni, O., Cafarella, M., Downey, D., Popescu, A.M., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Unsupervised named-entity extraction from the web: an experimental study. *Artificial Intelligence* **165**(1) (2005) 91–134
31. Katz, J., Fodor, J.: The structure of a semantic theory. *Language* (39) (1963) 170–210
32. Jackendoff, R.: *Semantics and Cognition*. MIT Press, Cambridge, MA (1983)
33. Saeed, J.: *Semantics*. 2 edn. Blackwell (2003)

BEATCA: Map-Based Intelligent Navigation in WWW

Mieczysław A. Kłopotek, Krzysztof Ciesielski, Dariusz Czerski,
Michał Dramiński, and Sławomir T. Wierchoń

Institute of Computer Science, Polish Academy of Sciences,
ul. Ordona 21, 01-237 Warszawa, Poland
{kłopotek, kciesiel, dcz, mdramins, stw}@ipipan.waw.pl

Abstract. In our research work, we explore the possibility to exploit incremental, navigational maps to build visual search-and-recommendation system. Multiple clustering algorithms may reveal distinct aspects of the document collection, just pointing to various possible meanings, and hence offer the user the opportunity to choose his/her own most appropriate perspective. We hope that such a system would become an important step on the way to information personalization. The paper presents the architectural design of our system.

Keywords: intelligent user interfaces, visualization, Web mining.

1 Introduction

Conveying the context of a returned document to a search engine user is one of challenging tasks for intelligent decision support systems. Within a broad stream of various novel approaches, we would like to concentrate on the well known 2-dimensional maps of document collections, as they are produced by WebSOM [16]. This means the semantics would be explained in terms of related documents. The drawback of WebSOM approach is to view the document collection from one perspective only.

The idea of information presentation in its context is not new. For a long time, already, the search engines reply to queries not only with simple ranked lists, but with labeled lists of groups of documents, that create contexts for individual documents. However, document maps seem to be the most appealing for humans with respect to provision of context awareness.

There have been several important projects in the recent years, concerned with map representation of document collections, just to mention *Themescape*¹, *SpaceCast*². A prominent position is taken by the *WebSOM* of Kohonen and co-workers [16]. However, the majority of systems makes the unrealistic assumption that document collection will not change in the future. A recent study described in [10] demonstrated deficiencies of various approaches to document

¹ <http://www.micropatent.com/static/index.htm>

² <http://www.geog.ucsb.edu/sara/html/research/spacecast/spacecast.html>

organization, including WebSOM, under non-stationary environment conditions of growing document quantity, proposing as a remedy dynamic self-organizing neural model DASH..

We took a different perspective in our own project, claiming that the adaptive and incremental nature of a document-map-based search engine cannot be confined to the map creation stage alone and in fact engages all the preceding stages of the whole document analysis process. We want to outline briefly our architecture beyond the map formation process.

The process of mapping a document collection to a two-dimensional map is a complex one and involves a number of steps which may be carried out in multiple variants. In our search engine BEATCA [3,4,5,6], the mapping process consists of the following stages (see Figure 2): (1) document crawling (2) indexing (3) topic identification, (4) document grouping, (5) group-to-map transformation, (6) map region identification (7) group and region labeling (8) visualization. At each of these stages various decisions can be made implying different views of the document collection, hence producing a different document map.

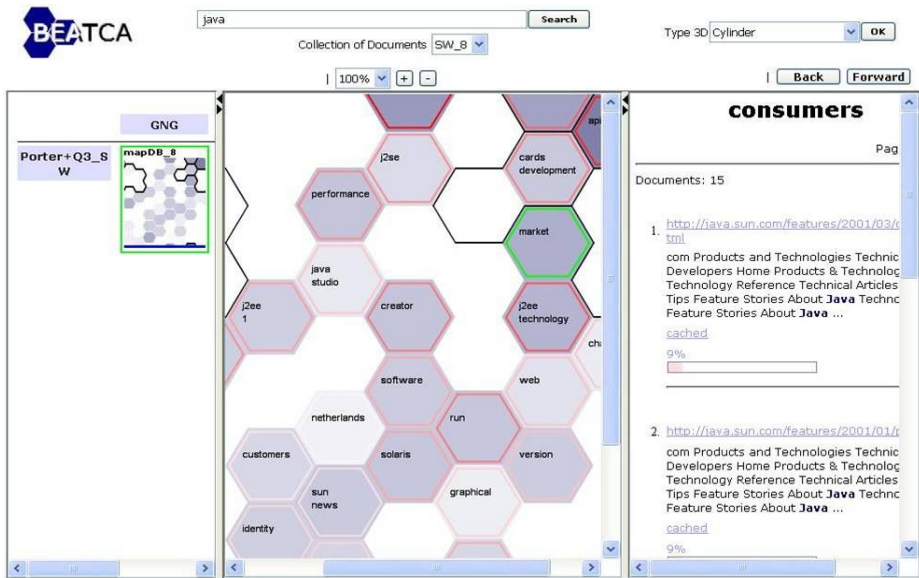


Fig. 1. BEATCA – user interface

For example, the indexing process involves dictionary optimization, which may reduce the documents collection dimensionality and restrict the subspace in which the original documents are placed. Topics identification establishes basic dimensions for the final map and may involve such techniques as SVD analysis [1], fast Bayesian network learning (ETC [11]) and other. Document grouping may involve various variants of growing neural gas (GNG) techniques, [8]. The group-to-map transformation, used in BEATCA, is based on SOM ideas, [16], but with

variations concerning dynamic mixing of local and global search, based on diverse measures of local convergence. The visualization involves 2D and 3D variants.

Due to strong parametrization, the user of BEATCA can accommodate the map creation process to his particular needs, or even generate multiple maps covering different aspects of a document collection.

The overall complexity of the map creation process, resulting in long run times, as well as the need to avoid "revolutionary" changes of the image of the whole document collection, require an incremental process of accommodation of new incoming documents into the collection.

Within the BEATCA project we have devoted much effort to enable such a gradual growth. The requirement of gradual growth imposed necessity of finding solutions both for the design of applied algorithms and for design of architecture. Thanks to this effort it became possible to achieve smooth vertical (new topics) and horizontal (new documents on current topics) growth of document collection without deterioration of map formation capability and map quality [7]

To ensure intrinsic incremental formation of the map, all the computation-intensive stages involved in the process of map formation (crawling, indexing, GNG clustering, SOM-clustering) need to be reformulated in terms of incremental growth.

In this paper we outline the general system architecture design considerations (section 2) and underline some peculiarities in map creation process (section 3) and visualization module (section 4) that were necessary to achieve the imposed goals.

2 General Architecture

Our research targets at creation of a full-fledged search engine (with working name BEATCA) for collections of up to a million documents, capable of representing on-line replies to queries in graphical form on a document map. We follow the general architecture for search engines, where the preparation of documents for retrieval is done by an indexer, which turns the HTML etc. representation of a document into a vector-space model representation, then the map creator is applied, turning the vector-space representation into a form appropriate for on-the-fly map generation, which is then used by the query processor responding to user's queries (see figure 2).

The architecture of our system has been designed to allow for experimental analysis of various approaches to document map creation. Software consists of essentially five types of modules, cooperating via common data structures. The types of modules are: (1) the robot, (spider, crawler) collecting documents for further processing, (2) indexer, transforming documents into vector space representation, (3) optimizer, transforming the document space dictionary into more concise form, (4) document clustering, identifying compact groups of documents sharing similar topics, (5) mapper, transforming the vector space representation into a map form (6) search engine, responding to user queries, displaying the document maps in response to user queries. On top of these modules, a

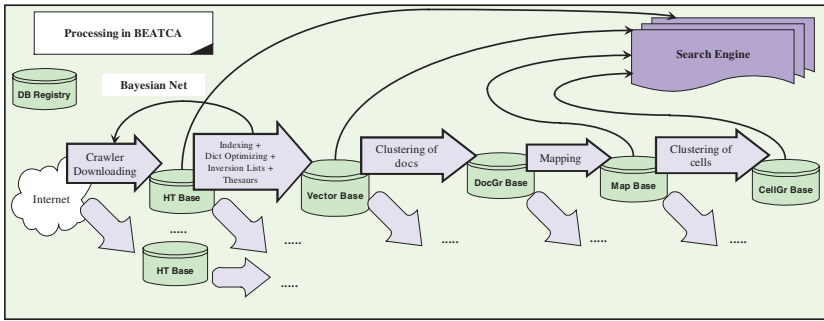


Fig. 2. BEATCA system architecture

special experiment oriented interface with a batch language interpreter (experiment manager) has been designed, which allows to run and re-run search engine processes, to select and compare variants of various components, measure their efficiency in terms of time and memory consumption as well as computing various quality measures of clustering and visualization.

To ensure interoperativity of various variants of modules achieving same goals, a decision was made that they will communicate via a database, creating and reading document collection related objects of well defined types.

The data structures, interfacing the modules, are of type: (1) HT Base [hypertext documents], (2) Vector Base [vector space representations], (3) DocGR Base [thematical document groups] (4) Map Base [repository of various maps], (5) CellGR Base [map areas (groups of cells)] (6) Base Registry [registry of all databases, parameters and evaluation results].

A HT Base is the result of a robot activity. We have currently two types of robots, one collecting documents from the local disk space, and another for the Web. A robot collects the hypertext files walking through links connecting them and stores them in a local directory and registers them in an SQL (currently MySQL) database. Standard information like download (update) date and time, original URL, summary (if extractable), document language and the list of links (together with information if already visited) is maintained by the robot.

A HT Base can be processed subsequently by an indexer and possibly an optimizer to form a Vector base for the document collection. A Vector Base is a representation of a document space the space spanned by the words (terms) from the dictionary where the points in space represent documents.

A Vector base is then transformed to a document map by a mapper process. A map is essentially a two-level clustering of documents: there are clusters of documents (stored in DocGR Base) and clusters of document clusters (stored in Map Base). Document clusters are assigned a graphical representation in terms of elementary "pixels" (labeled by appropriate phrases) in a visual representation, whereas clusters of document clusters are assigned "areas" consisting of "pixels". Note that in our approach we use a kind of multilevel maps, where higher levels "pixels" are "expanded" into maps/map fragments at a detailed level.

Note that the same HT Base may be processed by various indexers and optimizers so that out of a single HT Base many Vector bases may arise. Similarly one single Vector base may be processed by diverse mappers to form distinct maps. To keep track of the various descendants of the same HT Base, the Base Registry has been designed. The search engine makes use of all the maps representing the same HT Base choosing the one most appropriate for a given user query.

The search engine has been explicitly designed as a test-bed for various algorithmic solutions to constituent search engine components. Hence additional feature is a database keeping track of results of experiments (constituting of selections of process components and data sets as well as quality evaluation procedures). The database of experiments is filled (and used in case of continued experiments) by the experiment management module.

3 Mapper

One of main goals of this project is to create multidimensional document map in which geometrical vicinity would reflect conceptual closeness of documents in a given document set. Additional navigational information (based on hyperlinks between documents) is introduced to visualize directions and strength of between-group topical connections.

At the heart of the overall process is the issue of clustering documents. Clustering and content labeling is the crucial issue for understanding the two-dimensional map by the user. It has been recognized long time ago, that clustering techniques are vital in information retrieval on the Web [12]. We started our research with the WebSOM approach, but as our findings were similar to that of [10], that is both speed (one million of docs processed in several weeks, while we can do this in 3 days), clustering stability etc. were unsatisfactory for under non-stationary environment, that is one with incoming flow of new documents, with topic drift. The scalability of WebSOM approach is also discouraging (high complexity in collection size terms).

We guess that the basic problem with WebSOM lies in the initialization process of so-called reference vectors, being the centroids of the clusters to grow. In the original WebSOM they are initialized randomly, to be corrected later on in the clustering process. Such an initialization may lead to an instability during clustering, because the learning process of WebSOM possesses a "learning speed" parameter α , which may turn out to be too low to ensure convergence for a particular initialization. Another problem lies in the general concept of clustering. In WebSOM, it is tightly coupled with a (non-linear) projection from a multidimensional to a two-dimensional space. Now, there may be infinitely many such projections with equal rights. So one needs really a sense of goal for selecting the appropriate one.

The first issue we tackled was dictionary optimization strategies and their speed-up effects to tackle the complexity issue [3]. Another research direction was to obtain better clustering via fuzzy-set approach and immune-system-like clustering, [13]. Our approach to document clustering is a multi-stage one:

- clustering for identification of major topics (see [4,5])
- cellular document clustering (see [6])
- cellular document clusters to WebSOM map projection (see [3])
- cell clusters extraction and cell labelling (see [13])

In order to obtain a stable map, one needs to fix the perspective from which one looks at the document collection. This can be achieved if we identify major topics of the document collection. This is done in the step "clustering for identification of major topics". We suggest [6] a Bayesian approach, which was a result of our investigation of the behavior of the PLSA algorithm [9]. Alternatively, different initialization techniques could be used: in [5] we described an approach to major topic identification based on LSI/SVD, and in [7] we described usage of a version of Fuzzy-ISODATA algorithm for this purpose. Having identified the major topics, we can initialize the map in a more definite way, in particular imposing stabilization of the resulting maps [4].

After the topics have been identified, the documents need to be assigned to these and intermediate ones, and the relationships between the topics have to be identified. This process is called by us "Cellular document clustering", and leads to creation of a graph model of document collection. Three different techniques may be used at this point: the WebSOM (plain or hierarchical) approach, the GNG approach, or artificial immune systems (AIS) approach (see [15]).

The graph model of document collection needs to be visualized in form of a document map. Therefore, a step of "cellular document clusters to WebSOM map projection" is applied. It is surplus in case, when WebSOM (plain or hierarchical) is used to cluster documents, but is necessary for more elastic topologies like GNG model and AIS model to show the graph model in 2D or 3D (refer also to section 5).

Finally, for purposes of better readability of the document map, cells need to be joined into larger, topically uniform areas which is done in the step of "cell clusters extraction". Both cells and areas have to be labeled with descriptive terms or phrases [15].

4 Contextual Maps

In our work we use well known approach of representing documents as points in term vector space. It is a known phenomenon that text documents are not uniformly distributed over the space. Characteristics of frequency distributions of a particular term depend strongly on document location. On the basis of experimental results, we suggest to identify automatically groups containing similar documents as the preprocessing step in document maps formation. We argue that after splitting documents in such groups, term frequency distributions within each group become much easier to analyze. In particular, it appears to be much easier to select significant and insignificant terms for efficient calculation of similarity measures during map formation step. Such cluster hierarchy we call *contextual* groups. For each contextual group, separate maps are generated (Figure 3).

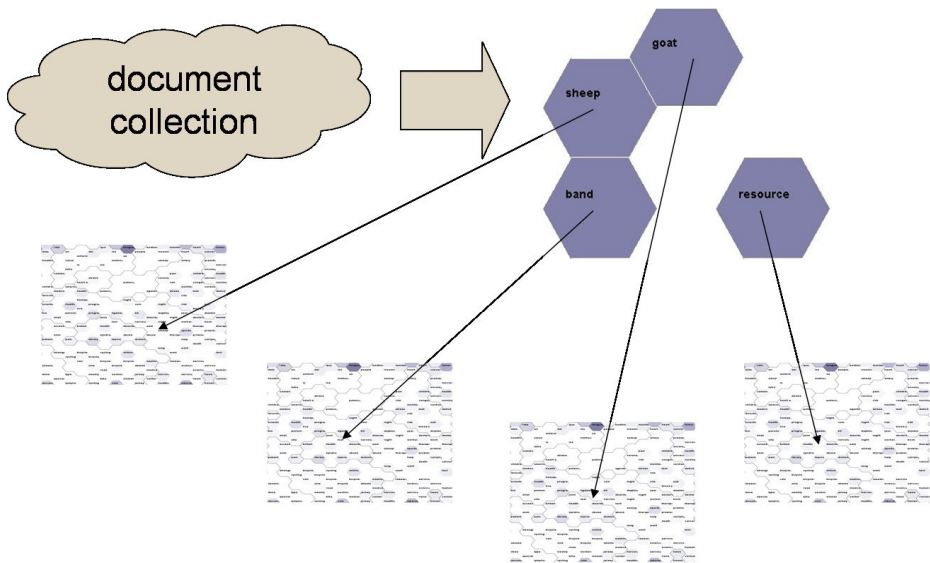


Fig. 3. Context-sensitive approach

Learning process of the contextual model is to some extent similar to the classic, non-contextual learning. However, the standard vector space representation is replaced with topic-sensitive term weighting, taking into account importance of the term within a particular thematic group. It should also be noted that each constituent contextual map can be processed independently, in particular learning process can be distributed. Also a partial incremental update of such models appears to be much easier to perform, both in terms of model quality, stability and time complexity (for detailed results see [7]). The hierarchy tree of the contextual maps is rooted at the map of general contexts (Figure 3).

5 User Interface

The presentation of document maps in our system is similar to the one used in the WebSOM project, but enriched with our own modifications.

There is a variety of modifications to the basic SOM topology, having different clustering and visualization properties. In particular, we have applied Euclidean SOMs with quadratic and hexagonal cells, projected on torus surface (figure 4) and presented in 2D.

At present, in our search engine map of documents can be presented by 2 dimensional map (see fig. 1), following WebSOM's paradigm [18], or in 3D, by cylinder (see fig. 5(a)), sphere (see fig. 5(b)) or torus (see fig. 4). Our basic concern was with boundary regions, where within the original WebSOM approach a too few neighbors phenomenon occurred. As suggested by WebSOM authors and others, we extended the map so that the left and the right boundaries



Fig. 4. 3D map – torus

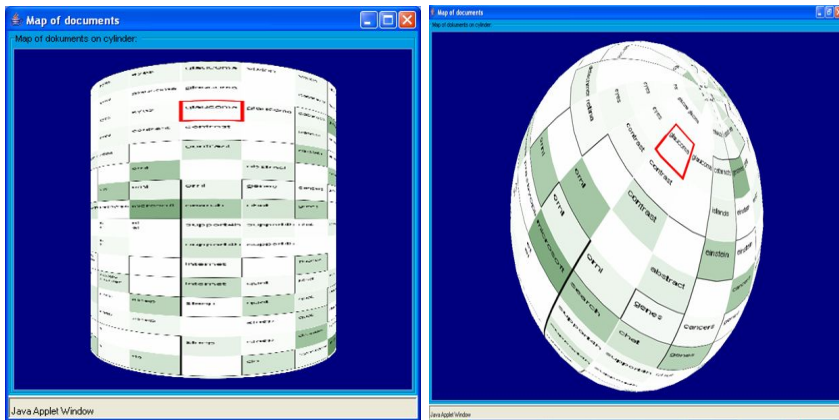


Fig. 5. 3D map – (a) cylinder (b) sphere

are connected. The same is with the top and the bottom one. As this junction is not visible in case of a planar map, it is visible with the rotating cylinder in the horizontal direction. Rotating sphere and torus representations make vertical junction visible also. The change in intended presentation has an impact of course on the way the clustering in WebSOM algorithm is implemented. Also clustering of map regions is affected by this concept. In case of sphere representation, also the distance measures in non-boundary regions are affected.

Our further extension concerns using a multi-map approach (maps generated with various clustering algorithms, hence explicating different aspects of the data). When presenting results of a query, the most appropriate map is selected

out of the ones available so that the documents in response to a query form "well-formed" clusters on such a map. The automated choice of the map is based on a criterion of split of response documents into several large clusters (scattering over the whole map and concentrating at a single spot are discouraged).

When a user enters a query, he may be advised on possibilities of query expansion, where additional terms are suggested by a Bayesian net based mechanism.

6 Concluding Remarks

As indicated e.g. in [10], most document clustering methods, including the original WebSOM, suffer from their inability to accommodate streams of new documents, especially such in which a drift, or even radical change of topic occurs.

The main contribution of this paper is to point at the possibility of design of visual search engines indexing millions of documents at acceptable speed, and providing multiple views of the same growing document collection. An appropriate architecture to achieve this goal has been developed. What distinguishes it from traditional approaches to document map creation is the addition of topical initialization, that stabilizes the map, decompositional "contextual maps", that permit to handle multiple subsets of documents in parallel, as well as focused crawling, that eases the tasks of clustering components by providing with topical portions of documents. Another feature absent from other systems is the possibility to have multiple modules performing the same function so that different variants of maps of the same document collection can be generated. An appropriate user query answering interface allows the user to select the map that best matches his needs, or let the system choose the map for him.

An added value is also the extension with an experiment manager which helps to identify bottlenecks to cope with.

The new (contextual) concept of document maps beyond our architecture leads to many interesting research issues, such as context-dependent dictionary reduction and keywords identification, topic-sensitive document summarization, subjective model visualization based on particular user's information requirements, dynamic adaptation of the document representation and local similarity measure computation, or even context-dependent inverse list construction. We plan to tackle these problems in our future works.

References

1. M.W. Berry, Z. Drmac, E.R. IJessup, Matrices, vector spaces and information retrieval, SIAM Review, Vol. 41, No. 2, pp. 335-362 1999
2. J. Chen, L. Sun, O.R. Zaiane, R. Goebel, Visualizing and Discovering Web Navigational Patterns, webdb2004.cs.columbia.edu/papers/1-3.pdf
3. K. Ciesielski, M. Draminski, M. Klopotek, M. Kujawiak, S. Wierzchon, Architecture for graphical maps of Web contents. Proc. WISIS'2004, Warsaw
4. K. Ciesielski, M. Draminski, M. Klopotek, M. Kujawiak, S. Wierzchon, Mapping document collections in non-standard geometries. B. De Beats et al. (eds): Current Issues in Data and Knowledge Engineering. Akademicka Oficyna Wydawnicza EXIT Publ., Warszawa 2004.. pp.122-132.

5. K. Ciesielski, M. Draminski, M. Kłopotek, M. Kujawiak, S. Wierzchon: Clustering medical and biomedical texts - document map based approach. Proc. Sztuczna Inteligencja w Inżynierii Biomedycznej SIIB'04, Krakw. ISBN-83-919051-5-2
6. K. Ciesielski, M. Draminski, M. Kłopotek, M. Kujawiak, S.T. Wierzchon, On some clustering algorithms for Document Maps Creation, in: Proceedings of the Intelligent Information Processing and Web Mining (IIS:IIPWM-2005), Gdansk, 2005
7. K. Ciesielski, M. Draminski, M. Kłopotek, D.Czerski, S.T. Wierzchon, Adaptive document maps, to appear in: Proc. IIPWM-2006, Ustroń, 2006
8. B. Fritzke, A growing neural gas network learns topologies, in: G. Tesauero, D.S. Touretzky, and T.K. Leen (Eds.) Advances in Neural Information Processing Systems 7, MIT Press Cambridge, MA, 1995, pp. 625-632.
9. T. Hoffmann, Probabilistic Latent Semantic Analysis, in: Proceedings of the 15th Conference on Uncertainty in AI, 1999
10. C. Hung, S. Wermter, A constructive and hierarchical self-organising model in a non-stationary environment, Int. Joint Conference in Neural Networks, 2005
11. M. Kłopotek, A new Bayesian tree learning method with reduced time and space complexity. Fundamenta Informaticae, 49(no 4)2002, IOS Press, pp. 349-367
12. M. Kłopotek, Intelligent information retrieval on the Web. in: Szczepaniak, Piotr S.; Segovia, Javier; Kacprzyk, Janusz; Zadeh, Lotfi A. (Eds.): (2003) Intelligent Exploration of the Web Springer-Verlag ISBN 3-7908-1529-2, pp. 57-73
13. M. Kłopotek, M. Draminski, K. Ciesielski, M. Kujawiak, S.T. Wierzchon, Mining document maps, in Proceedings of Statistical Approaches to Web Mining Workshop (SAWM) at PKDD'04, M. Gori, M. Celi, M. Nanni eds., Pisa, 2004, pp.87-98
14. M. Kłopotek, S. Wierzchon, K. Ciesielski, M. Draminski, D. Czerski, M. Kujawiak, Understanding nature of map representation of document collections map quality measurements . Proc. Int. Conf. Artificial Intelligence Siedlce, September 2005.
15. M. Kłopotek, S. Wierzchon, K. Ciesielski, M. Draminski, D. Czerski, Conceptual maps and intelligent navigation in document space (in Polish), to appear in: Akademicka Oficyna Wydawnicza EXIT Publishing, Warszawa, 2006
16. T. Kohonen, Self-Organizing Maps, Springer Series in Information Sciences, vol. 30, Springer, Berlin, Heidelberg, New York, 2001
17. P. Koikkalainen, E. Oja, Self-organizing hierarchical feature maps, in Proc. International Joint Conference on Neural Networks, San Diego, CA 1990 501 pages. ISBN 3-540-67921-9, ISSN 0720-678X
18. K. Lagus, Text Mining with WebSOM, PhD Thesis, Helsinki Univ. of Techn., 2000
19. A. Rauber, Cluster Visualization in Unsupervised Neural Networks. Diplomarbeit, Technische Universität Wien, Austria, 1996
20. J.A. Wise, J. Thomas, J., K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: Spatial analysis and interaction with information from text documents, in IEEE Information Visualization. 1995. p. 51-58.
21. A.H. Youssefi, D.J. Duke, M.J. Zaki, Visual Web Mining, <http://www2004.org/proceedings/docs/2p394.pdf>, WWW2004, May 1722, 2004, New York, NY USA.

Model-Based Monitoring and Diagnosis Chip for Embedded Systems

Satoshi Hiratsuka, Hsin-Hung Lu, and Akira Fusaoka

Department of Human and Computer Intelligence
Ritsumeikan University
Nojihigashi, Kusatsu-city, SIGA, Japan 525-8577
fusaoka@ci.ritsumeai.ac.jp

Abstract. In this paper, we propose a design consideration for a monitoring and diagnosing chip for the embedded system based on the model-based diagnosis. We introduce the qualitative model for the embedded system by transforming the continuous dynamics of components into the discrete state transition system, which is then further transformed into the circuit called Synchronous Boolean Network(**SBN**). The faults of system components are reduced to the stuck-at faults in **SBN**. We present a hardwired **SBN** diagnosis engine based on Roth's D-calculus, which allows efficient identification of the faulty parts by propagating the anomaly through the **SBN** structure.

1 Introduction

In this paper, we present a design consideration for an on-line monitoring and diagnosis chip for the embedded system based on the model-based diagnosis. A dynamical system is called “embedded” when it contains the continuous dynamics managed by the discrete control. Most industrial systems today are embedded because they are usually monitored by the embedded microprocessors. A large parts of the monitor system are occupied by the programs for on-line diagnosis. For example, 75% of software module in the Electronic Control Units (ECUs) of automobiles deal with the detection and recovery for the faults in the engine system [2]. Therefore, an efficient and high-responsible diagnosis method is one of the most intriguing and significant issues for the embedded system. A model-based diagnosis (hereafter **MBD**) has been widely investigated for embedded systems and generated a lot of practical diagnosis engines. These systems have been mostly applied to the off-line and off-board diagnosis. For a safety critical system such as automobiles, however, an on-board diagnosis is one of the best approaches to achieve the high responsible diagnosis. But there are some difficulties in the on-board implementation of **MBD** due to the high complexity computation (NP-hard) and the hardware limitation (the sensors are sparse and the memory size is limited). The fundamental method of **MBD** is the consistency-based diagnosis which iterates the checking whether the behavior predicted from the model is consistent with the observation. So that it requires the deduction or simulation to predict the correct behavior from the model. Struss proposed the

state-based diagnosis as one of the promising approach to the on-board **MBD** [9]. In the state-based diagnosis, the fault is detected by checking whether the observed state exists in the set of possible states so that the time consuming simulation can be avoided. Taking another approach, Casio et. al. proposed a compilation-based methodology in which an off-line model-based diagnosis is transformed into a decision tree for on-board implementation [2]. Although the decision tree allows a compact representation of diagnosis procedure, it is sometimes insufficient due to the lack of ability to deal with the temporal behavior of the system. Console, et.al. improve the compilation approach by introducing the temporal decision tree [3].

Contrary to the both approaches which aim at the reduction of on-board computation, we propose in this paper a straightforward implementation of the consistency-based diagnosis in the hardware logic in which the on-board simulation and the consistency checking are embedded in the form of LSI logic. We introduce two stages of abstraction to generate the model. At the 1st stage, the qualitative model for the embedded system are formed by transforming the continuous dynamics of components into the discrete state transition system. At the 2nd stage, it is transformed again into circuit called Synchronous Boolean Network(**SBN**) . The **SBN** is a network of the Boolean gates with feedback, which simulates and monitors the dynamics of the system by comparing the observables with the simulation result at every time-step. The faults of system components are reduced to the more comprehensible stuck-at faults in **SBN**. The permanent fault of the component is easily reduced to the stuck-at faults of **SBN**. A stuck-at fault means that the input or output of the faulty element is fixed to be 0 or 1. The model-based diagnostic algorithms have been studied for the stuck-at fault of **SBN** [6]. Here we present a hardware implementation of this algorithm, which allows the incremental conflict set formation by propagating the anomaly through the **SBN** structure.

2 Model-Based Diagnosis and Model Abstraction

In the **MBD**, the model is defined by the system description, which is an ensemble of logical formulas of the form $Ok_i \supset F_i$ for each component i . The assumable Ok_i is used to denote the health of the component _{i} and F_i describes its dynamics. The interconnection of the components is naturally represented by the sharing of variables. From the system description SD and the symptom OBS , the diagnostic engine generates the conflict sets $(\neg Ok_1 \vee \neg Ok_2 \vee \dots)$ such that $SD \wedge OBS \vdash (\neg Ok_1 \vee \neg Ok_2 \vee \dots)$. This means at least one of $\{\text{component}_1, \text{component}_2, \dots\}$ must be fault as the cause of the symptom [4]. A set of the minimal conflict sets forms the diagnosis.

Definition of System Description

A system description is $SD = (V, OK, Comp)$ where

- (1) V is a set of variables for the whole system such that

Input variables: $SI \subseteq V$; Output variables: $SO \subseteq V$; Observables: $SB \subseteq V$

- (2) OK is a set of predicates Ok_i which means component i is healthy.
- (3) $Comp$ is a set of component description $C_i = (X_i, Y_i, F_i)$ such that
 - $X_i \subseteq V$: Input variables of the component i
 - $Y_i \subseteq V$: Output variables of the component i
 - $F_i \subseteq X_i \times Y_i$: Causal relations of the component i such that $Ok_i \supset Y_i = F_i(X_i)$

The system description represents the connectivity between components by the sharing of variables so that the following connectivity condition must be satisfied.

$$\forall i[\forall x \in X_i \exists Y_j[x \in Y_j \vee x \in SI]]; \quad \forall j[\forall y \in Y_j \exists X_i[y \in X_i \vee y \in SO]]$$

Since the abnormal behavior of the system is qualitative in nature, we need not necessarily the precise description of continuous dynamics in the diagnosis model if it contains the sufficient description for the causal relation between the faulty of components and the anomaly of the system. Namely, the granularity of the model description depends on the underlying fault ontology of the actual diagnosis. The model abstraction plays an essential role to reduce the inherent complexity of diagnosis.

Definition of Model Abstraction

For any two system descriptions $SD = (V, OK, Comp)$, $SD' = (V', OK', Comp')$, S' is a model abstraction of S if and only if there exists a mapping $\alpha : V \rightarrow V'$ which satisfies the following two conditions

- (1) Simulation Condition

$$\forall C_i(X_i, Y_i, F_i) \in Comp \exists C'_j(X'_i, Y'_i, F'_i) \in Comp' [Y_i = F(X_i) \supset \alpha(Y_i) = F'_i(\alpha(X_i))]$$
- (2) Preservation of Ok -ness

$$\forall Ok_i \in OK \exists Ok'_j \in OK' [Ok_i \equiv Ok_j] \text{ and vice versa}$$

The simulation condition is necessary for the on-line monitoring. The anomaly of system behavior is detected when the discrepancy:

$$X'_i = \alpha(X_i) \wedge Y'_i \neq \alpha(Y_i)$$

is found. On the other hand, it is required to have an one-to-one correspondence between Ok_i and Ok'_i .

In the following sections, we will use two stages of abstraction. In the first stage, we will map the continuous dynamics of the system into a qualitative model with sufficient granularity to detect the anomaly. In the second stage, we will then map the qualitative model into its circuit representation(SBN).

3 Qualitative Fault Model

3.1 A Single Cylinder

Throughout this paper, we use a single cylinder system for explanation. A single cylinder system (Fig.1) is a simplified model of the automotive engine which is

composed of a cylinder, a crankshaft, an intake valve, an exhaust valve and a spark ignition actuator. The cylinder system burns the fuel and generates the torque by repeating the following cylinder cycle (Fig.2):

- (1) I-state (Intake): the mixture of fuel and air m is entered into the cylinder. Its quantity(mass) depends on the situation of the intake valve v , which is determined by **ECU** at the start of this state. The position of the piston $p(t)$ is moved from the top to the bottom of the cylinder during this state.
- (2) C-state (Compression): The air/fuel mixture is compressed. The piston is going up from the bottom to the top.
- (3) E-state (Expansion): The spark ignites I at the start of this state. The pressure inside the cylinder is rapidly increased due to the flaming and explosion; in this case, the piston moves down from the top to bottom so that the torque $T(t)$ is generated and then transmitted to the crankshaft via the connection rod. Finally, the piston moves down to the bottom.
- (4) H-state (Exhaust): the exhaust gas g is expelled. The position of the piston is moved from the bottom to the top of the cylinder.

The cylinder system is a typical example of the hybrid system [1], since its dynamics is composed of the discrete state transition of the cylinder cycle and the continuous change of the physical quantities occurs only in each state which is governed by the (differential) equations given in the state. The dynamics of the whole system can be described by the differential equation about the crankshaft revolution speed n . Note that the torque is generated only in the E-state so that n is gradually decreasing due to the load torque in any other state. Namely,

$$\dot{n}(t) = an(t) (a < 0) \text{ in state I,C,H,} \quad \dot{n}(t) = an(t) + bT(t) \text{ in state E}$$

The position of the piston is represented by the angle of the crankshaft so that its deviation is in proportion to n : $\dot{p}(t) = cn(t)$.

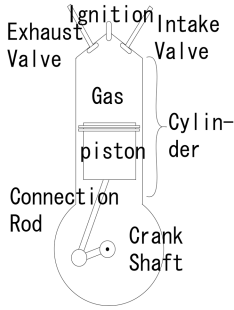
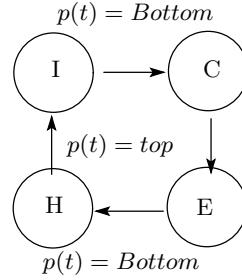
We assume that the generated torque is in proportion to the mass m of the inflamed air/fuel mixture which is determined by the situation of valve:

$$T(t) = km(t), m(t) = rv(t), \quad \text{where } k, r \text{ are constants}$$

We assume that $n(t)$ and the temperature J in the cylinder are observed at the end of each cylinder cycle. The valve $v(t)$ of the air-fuel mixture is controlled at the beginning of Intake state depending on $n(t)$. Any other physical parameters are not observable.

3.2 Qualitative Model

We introduce the qualitative model of the cylinder as the first stage of abstraction. Since the change of the crankshaft revolution speed n is considered to be very small within one cylinder cycle, we can regard n as constant at each step of diagnosis. In Table 1, we present the discretization of the physical quantities. The value of $\dot{n}(t)$ is divided into 8 levels $L_0 < L_1 < L_2 < L_3 < L_4 < L_5 < L_6 < L_7$ and the value of other physical parameters are divided into two values: L (low) and H (high).

**Fig. 1.** The cylinder system**Fig. 2.** The cylinder cycle**Table 1.**

physical Parameter	Variable	Landmark value
Temperature	J	$H(\text{high}), L(\text{low})$
Valve	V	$H(\text{full open}), L(\text{half open})$
Mixture Gas	m	$H(\text{large amount}), L(\text{small amount})$
Remaining Exhaust Gas	g	$(H(\text{large amount}), L(\text{small amount}))$
Acceleration of Shaft	\dot{n}	$L_0 < L_1 < L_2 < L_3 < L_4 < L_5 < L_6 < L_7$
Torque	T	$H(\text{big}), L(\text{small})$
Ignition	Ig	$H(\text{fired}), L(\text{misfired})$

We transform the cylinder cycle into the discrete event transition by replacing the continuous dynamics of physical variables in each state with the corresponding landmark values at the end of the state. This abstraction is feasible if we can recognize the abnormal behavior occurred in the state from the observation at the end of the state or the cylinder cycle. Namely, we assume that every observable has a fixed value at the start or the end of state which can be estimated by monitoring. By this abstraction, we can describe the discrete dynamics by the form

φ at the beginning of state $\supset \psi$ at the end of state

We simply write this form as $\varphi \supset \circ\psi$, where \circ is the next operator which means the predicate becomes true after the time delay of the quarter of the cylinder cycle. The differential equations in each state is reduced into the following qualitative difference equations, which comprises the system description of the first abstraction model.

- (1) **state I:** If the valve is not opened completely, the mass of fuel will not be full; $v = L \supset \circ(m = L)$.

If the exhaust gas is not expelled at the end of the previous state H, the fuel will not be full even if the valve is completely open; this suggests the existence of the fault of the valve for the exhaust gas; $v = H \wedge g = H \supset \circ(m = L)$.

Otherwise, the fuel will be full; $v = H \wedge g = L \supset \circ(m = H)$.

\dot{n} is gradually decreasing (we assume one step down);

$\dot{n} = L_x \supset \circ(\dot{n} = L_{x-1}); \dot{n} = L_0 \supset \circ(\dot{n} = L_0)$; where $x \in \{1, \dots, 7\}$;

- (2) **state C:** In this state, the illegal ignition may occur due to the excessively high temperature. It causes the drastic decreasing of \dot{n} (3 steps down);

$\dot{n} = L_x \wedge J = H \supset \circ(\dot{n} = L_{x-3})$, where $x \in \{1, \dots, 7\}$.

In normal case, \dot{n} is gradually decreasing; $\dot{n} = L_x \wedge J = L \supset \circ(\dot{n} = L_{x-1})$.

If the rotation speed is the lowest, it keep the value; $\dot{n} = L_0 \supset \circ(\dot{n} = L_0)$.

- (3) **state E:** When the temperature is low, the ignition occurs correctly;

$J = L \supset Ig = H$,

and the torque is generated in proportion to the mass of air-fuel mixture in the cylinder; $Ig = H \wedge m = X \supset \circ(T = X)$ where $X = L, H$.

When generated torque is high, \dot{n} is going up to the highest level L_7 else it is going up to L_5 ; $T = H \supset \circ(\dot{n} = L_7); T = L \supset \circ(\dot{n} = L_5)$.

If the ignition occurred already at the previous state C due to high temperature, it does not fire again. In this case, the torque is not generated;

$J = H \supset Ig = L$.

In this case, \dot{n} is gradually decreased.

$Ig = L \wedge \dot{n} = L_x \supset \circ(\dot{n} = L_{x-1}); \dot{n} = L_0 \supset \circ(\dot{n} = L_0)$; where $x \in \{1, \dots, 7\}$.

- (4) **state H:** The \dot{n} is gradually decreased in this state.

$\dot{n} = L_x \supset \circ(\dot{n} = L_{x-1}); \dot{n} = L_0 \supset \circ(\dot{n} = L_0)$; where $x \in \{1, \dots, 7\}$.

- (5) **Control rules:** We assume that the control objective is to achieve the stability of n . In order to keep the value of n stable (around L_6), the control rule is selected according to the value of \dot{n} at the end of the state H; $\dot{n} < L_6 \supset \circ(v = H); \dot{n} \geq L_6 \supset \circ(v = L)$.

3.3 SBN Model

We introduce the SBN model as the second stage of abstraction. In the above formulas, all predicates φ, ψ in $(\varphi \supset \circ\psi)$ have the form $X = U$ where X is a physical parameter and U is the landmark value of the quantification. $X = U_1$ and $X = U_2$ are always exclusive if $U_1 \neq U_2$ so that we can simplify the description by introducing Boolean variables to encode these predicates. The encoding would result in the following Boolean formulas of (temporal) logic.

- (1) I state
 $\circ N_2 = N_2(N_0 + N_1), \circ N_1 = N_1 N_0 + N_2 \overline{(N_1 + N_0)}, \circ N_0 = (N_1 + N_2) \bar{N}_0,$
 $\circ M = \bar{G}V$
- (2) C state
 $\circ N_2 = N_2(N_1 + N_2)(N_1 + \bar{J}), \circ N_0 = (N_0 + \bar{J})(\bar{N}_0 + J)(N_1 + N_2),$
 $\circ N_1 = (N_1 + N_2)(\overline{N_0 + \bar{J}} + N_2)(\bar{N}_0 + \bar{J})(\bar{N}_0 + J + N_1)$
- (3) E state
 $\circ N_2 = Ig + N_2(N_1 + N_0), \circ N_0 = Ig + \bar{N}_0(N_1 + N_2), \circ V = \bar{N}_1 + \bar{N}_2,$
 $\circ N_1 = IgM + \bar{I}_g(N_1 + N_2)(N_1 + \overline{N_1 + N_0})(N_0 + \overline{N_1 + N_0})$
- (4) H state
 $\circ N_2 = N_2(N_0 + N_1), \circ N_1 = N_1 N_0 + N_2 \overline{(N_1 + N_0)}, \circ N_0 = (N_1 + N_2) \bar{N}_0$

From these formulas, we can design the circuit representation of **SBN**(Fig.3), which provides the system description in this stage of abstraction. In Fig.3, each of 4 large blocks encircled in dotted lines corresponds to each state of a cylinder cycle. The clocked latches are placed at the output of each block in order to synchronize them to the piston cycle. This circuit allows the on-board simulation to generate behavior of the cylinder. The fault of components can be found when the discrepancy occurs between the result of simulation and the observation. We can regard the permanent fault of a component as the set of events in which the physical parameters are frozen at some fixed value. In the **SBN** model, these events are represented by the stuck-at fault of the corresponding Boolean variables.

In Fig.3, the stuck-at faults occur only at the following lines:

- (1) The line 1,2,3 (stuck-at 0); the fault of the connection rod.
- (2) The line 11 (stuck-at 1); the fault about the abnormal temperature.
- (3) The line 12 (stuck-at 0); the fault of the valve for the air-fuel mixture.
- (4) The line 13 (stuck-at 0); the fault of the valve for the exhaust gas.

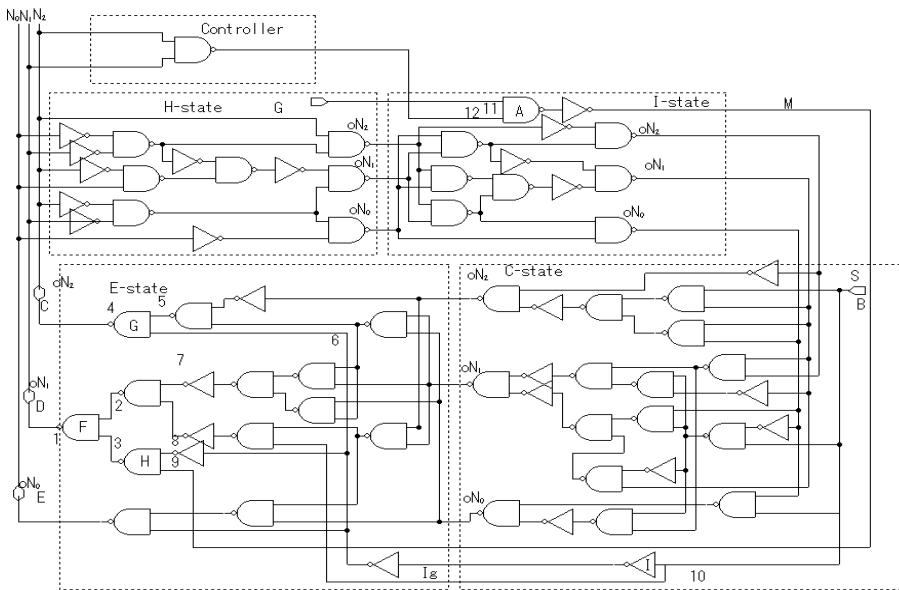


Fig. 3. SBN model of the single cylinder system

4 Diagnostic Engine

4.1 Roth' D-Calculus

In order to find out the stuck-at fault efficiently, we will apply the classical Roth' D-calculus [7] to the stuck-at fault. D-calculus is one of the 4-valued logic with

two new values D , and \bar{D} . $X = D$ means that X is forced to be $F(false)$ though it should be $T(true)$ if the whole system works correctly. Similarly, $X = \bar{D}$ means X is illegally T . Values D is propagated through Boolean operation. For example, $(X = T) \wedge (Y = D)$ contains the value D . In addition to the use of D-calculus, we use two types of gates in **SBN**. One is the *normal gate* which always operates correctly so that it only propagates the faults. Another is the *critical gate* which is corresponding to the critical part of the system so that it may cause the stuck-at faults. The gate of this type does not only propagate the existing conflict set but expands it by adding the new candidate of faults. In Table 2, we give the operations of NAND gate with L/R input and Y output in D-calculus. These operations can be executed by the circuits given in Fig.4. We call these circuit for NAND gate “R-gate”.

Table 2. Truth table of 4-valued propositional logic

		Normal gate				Critical gate			
		Y				Y			
$L \backslash R$		T	F	D	\bar{D}	T	F	D	\bar{D}
T		F	T	\bar{D}	D	\bar{D}	D	\bar{D}	D
F		T	T	T	T	D	T	T	D
D		\bar{D}	T	\bar{D}	T	\bar{D}	T	\bar{D}	D
\bar{D}		D	T	T	D	D	D	D	D

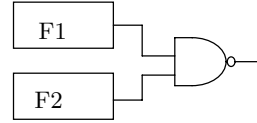


Fig. 4. Circuit Description

4.2 The Diagnostic Model

In the previous section, we have generated the **SBN** model(Fig.3) to simulate the correct behavior of the single cylinder system. We call this circuit **SBN-s**. The outputs of **SBN-s** are used for monitoring by being compared with the observed values of the cylinder. The permanent faults of the components are reduced into the stuck-at faults of the corresponding gate (critical gate) in **SBN-s**. In order to design the diagnostic engine in hardware, we construct another circuit called **SBN-d** by replacing each gate of the **SBN-s** with the corresponding R-gates. Namely, **SBN-d** simulates the system dynamics even in the faulty case by propagating D values.

4.3 The Conflict Set Formation

From the single faulty assumption, we can assume that **SBN-s** has at most one stuck-at fault. For each critical gate, we introduce the *OK* variables $L0, L1$ (stuck-at fault 0, 1 for L input) and $R0, R1$ (stuck-at fault 0, 1 for R input). A conflict set is represented by the set of these *OK* variables.

We can compute the all minimal conflict sets by starting from the conflict-set of the lower gates of input-side in **SBN-d** and by propagating and expanding it

to the higher level gates. We have proposed a method to generate the conflict set from those of the parts of the circuit in the following [6].

Let $F(X, Y)$ be a combinational circuit with the inputs X and the output Y . For the given input values X , the value of the output Y is correct if X coincides with the value of Y which is generated when F is a fault-free circuit. The minimal conflict set is empty if the output Y is correct. On the other hand, we always have non-empty conflict set for the incorrect output. Assume that F consists of parts F_1, F_2 and a NAND gate called key-gate (Fig.4). We have 32 possible cases of I/O values (16 for the normal gate and the other 16 for critical gate in the Table 2). We denote a conflict set of F_1, F_2, F by A, B, C , respectively. Whether $Y = T$ or F , C is empty so that A, B must be clear if they are not empty. If the key gate is normal, C is formed from only A, B . On the other hand, the OK variables of the key-gate ($L0, L1, R0, R1$) are added to A, B in the case of the critical gate. Let assume that $X = D, D$ and $Y = \bar{D}$. Since F_1 and F_2 are both incorrect output 0, the key gate will finally generate incorrect output 1. The minimal conflict set for F can be formed from those of F_1 and F_2 in the following manner.

When key-gate is normal, the OK variables in both A, B is still the candidates for the faults of F so that $C = A \cup B$. On the other hand, if the key-gate is critical, the inputs line of the key-gate itself may be faulty (stuck-at 0) so that $C = A \cup B \cup \{L0, R0\}$. Some other cases also show a similar behavior, and here we will show the conflict set formation rules for all cases in Table 3.

Table 3. Conflict generation and propagation

	Normal gate				Critical gate			
	Y				Y			
$L \setminus R$	T	F	D	\bar{D}	T	F	D	\bar{D}
T	ϕ	ϕ	B	B	$\{L0, R0\}$	$\{R1\}$	$\{L0, R0\} \cup B$	$\{R1\} \cup B$
F	ϕ	ϕ	ϕ	ϕ	$\{L1\}$	ϕ	ϕ	$\{L1\} \cap B$
D	$\{A\}$	ϕ	$A \cup B$	A	$\{L0, R0\} \cup A$	ϕ	$\{L0, R0\} \cup A \cup B$	$\{L1\} \cap B$
\bar{D}	A	ϕ	B	$A \cap B$	$\{L1\} \cup A$	$\{R1\} \cap A$	$\{R1\} \cap A$	$A \cap B$

4.4 Schematic Architecture of the Chip

We present a schematic architecture in Fig.6. It consists of two parts: the monitoring and diagnostic engine. The estimated size of the chip is not exceeding a few thousand gates for the qualitative model in the sect.3.

5 Concluding Remarks

In this paper, we reduce the on-line diagnosis of the embedded system into the stuck-at fault detection of **SBN** and present a chip configuration of the diagnostic engine. Comparing to other approaches, our method has an advantage that it allows the on-board simulation so that the consistency checking can be

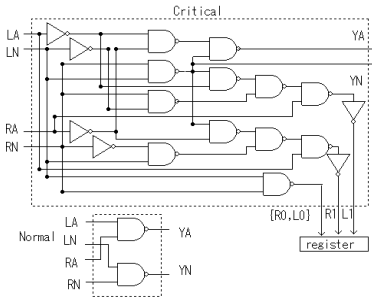


Fig. 5. R-gate for critical and normal gate

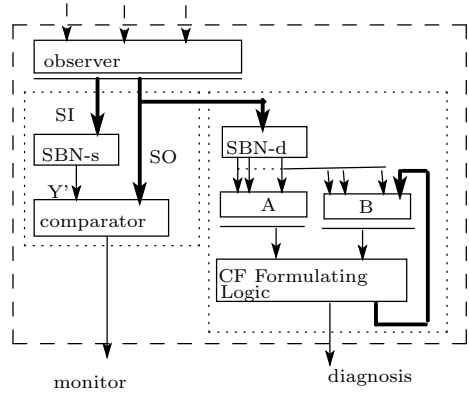


Fig. 6. Schematic chip architecture

performed in the real-time situation at the sacrifice of the additional hardware. For the faults of the embedded system, the number of critical gates (the gate with the possibility of fault) seems to be sparse, so that hardware implementation is possible within a moderate size of circuit.

References

1. Balluchi, A., Natale, F.D., Sangiovanni-Vincentelli, A. and van Schuppen, J.H., 2004. Synthesis for Idle Speed Control of an Automotive Engine, *Proc. of 7th International Workshop on Hybrid Systems: Computation and Control, HSCC2004*, LNCS 2993, 80-94.
2. Console, L., Picardi, C. and Dupre, D.T., 2003. Temporal Decision Tree: Model-based Diagnosis of Dynamic System On-Board, *Journal of Artificial Intelligence Research*, 19, 469-512.
3. Cascio, F., Console, L., Guagliumi, M., Osella, M., Panati, A., Sottano, S. and Dupre, D.T., 1999. Generating on-board diagnosis of dynamic automotive systems based on qualitative models, *AI Communications*, 12(1), 33-43.
4. Darwiche, A., 1998. Model-based diagnosis using structured system descriptions, *Journal of Artificial Intelligence Research*, vol 8, 165-222.
5. Dressler, O., 1996. On-Line Diagnosis and Monitoring of Dynamic Systems based on Qualitative Models and Dependency-recording Diagnosis Engines, *Proc. of 12th European Conference on Artificial Intelligence*, 461-465.
6. Hiratsuka, S., and Fusaoka, A., 2000. On a Model-based Diagnosis for Synchronous Boolean Network, *Proc. 13th IEA/AIE 2000*, LNAI-1821, Springer. 198- 203
7. Roth, J.P., 1966., Diagnosis of Automata Failures: A Calculus and a Methods, *IBM Journal of Research and Development* 10:278-291.
8. Sachenbacher, M., Struss, P. and Calen, C.M., 2000., A prototype for model-based on-board diagnosis of automotive systems, *AI Communications*, 13(2), 83-97. *IEEE trans. on Automatic Control*, Vol.40, NO.9, 1555-1575.
9. Struss, P. 1997., Fundamentals of Model-based Diagnosis of Dynamic Systems, *Proc. 15th International Conf. on Artificial Intelligence* 480-485.

A Knowledge-Based Approach for Automatic Generation of Summaries of Behavior

Martin Molina and Victor Flores

Department of Artificial Intelligence, Universidad Politécnica de Madrid,
Campus de Montegancedo S/N 28660 Boadilla del Monte, Madrid, Spain
{mmolina, vflores}@fi.upm.es

Abstract. Effective automatic summarization usually requires simulating human reasoning such as abstraction or relevance reasoning. In this paper we describe a solution for this type of reasoning in the particular case of surveillance of the behavior of a dynamic system using sensor data. The paper first presents the approach describing the required type of knowledge with a possible representation. This includes knowledge about the system structure, behavior, interpretation and saliency. Then, the paper shows the inference algorithm to produce a summarization tree based on the exploitation of the physical characteristics of the system. The paper illustrates how the method is used in the context of automatic generation of summaries of behavior in an application for basin surveillance in the presence of river floods.

1 Introduction

General techniques for automatic summarization usually simulate human reasoning such as abstraction or relevance reasoning. For example, techniques for event summarization include exploiting the saliency of events (with domain properties or statistics), abstracting events from collections of events, and integrating events based on semantic relations [1]. A particular application of automatic summarization is report generation in the context of control centers where the behavior of a dynamic system is supervised by human operators. Here, operators make decisions on real-time about control actions to be done in order to keep the system behavior within certain desired limits according to a general management strategy. Examples of these dynamic systems are: a road traffic network, the refrigeration system of a nuclear plant, a river basin, etc.

In this context, physical properties of dynamic systems provide specific criteria to formulate more specific techniques for summarizing and relevance reasoning. According to this, we present in this paper a knowledge-based approach that can be used to generate summaries in the context of surveillance of the behavior of dynamic systems. In the paper we analyze the type of knowledge and representation required for this type of task and we describe the main steps of an inference algorithm. We illustrate this proposal with the case of a particular application in the field of hydrology where thousands of values are summarized in single relevant states. At the end of the paper we make a comparative discussion with similar approaches.

2 The Method for Summarization

In automatic summarization two separated tasks can be considered: (1) *summarize* the most important information (i.e., *what* to inform) and (2) *present* the information using an adequate communication media according to the type of end-user (*how* to present the information). This paper describes our approach for the summarization task and, then, the paper illustrates how it is related to the presentation task in a hydrologic domain.

According to modern knowledge engineering methodologies [2], we have designed a method conceived with a set of general inference steps that use domain specific knowledge. In the following, we first describe the types of domain knowledge used in the method: (1) *system model*, (2) *interpretation model* and (3) *salience model*. Then, we describe the general inference as an algorithm that uses these models with a particular control regime.

2.1 The System Model

The *system model* is a representation of an abstraction about behavior and structure of the dynamic system. Our method was designed to simulate professional human operators in control centers with partial and approximated knowledge about the dynamic system. Therefore, the system model was conceived to be formulated with a qualitative approach instead of a precise mathematical representation with quantitative parameters.

The representation for the system model is a particular adaptation of representations and ontologies used in qualitative physics (e.g., [3] [4] [5] [6]). In the model, a detailed hierarchical representation of the structure is followed to support summarization procedures. However, since the system model is not used for simulation of the dynamic system, the behavior is represented with a simpler approach.

In particular, the structure of the dynamic system is represented with a set of *components* $C = \{C_i\}$. Each component represents a physical object of the system such as a reservoir, river or rainfall area in the basin. In a given moment, a component C_i presents a qualitative *state*. Each component C_i is also characterized in more detail with quantitative measures corresponding to physical *quantities* Q_1, \dots, Q_k (e.g., water-level and volume of a reservoir). Components are related to other components with the relations *is-a* and *member* (user-defined relations can also be used to consider domain-specific relations). A *parameter* is a tuple $P_i = \langle C_i, Q_i, F_i, T_i \rangle$ that represents a physical variable defined by the component C_i , the quantity Q_i , optionally a *function* F_i (e.g., as average time value, time derivative, maximum value, etc.) and optionally a *temporal reference* T_i (temporal references are *time points* or *time intervals*). An example of parameter is $\langle \text{Casasola}, \text{level}, \text{max}, [18:00, 21:00] \rangle$ which means the maximum value of the water level in the Casasola reservoir between 18:00 and 21:00 hours.

The model includes also a simplified view of the system behavior represented with *causal relations* between physical quantities. These relations can include labels such as temporal references about delay or type of influence (M^+ or M^- , i.e. increasing or decreasing monotonic functions, etc.). *Historical values* also help to represent information about behavior (e.g., average values, maximum historical values, etc.). Figure 1 shows a simplified example in the hydrologic domain that summarizes this representation.

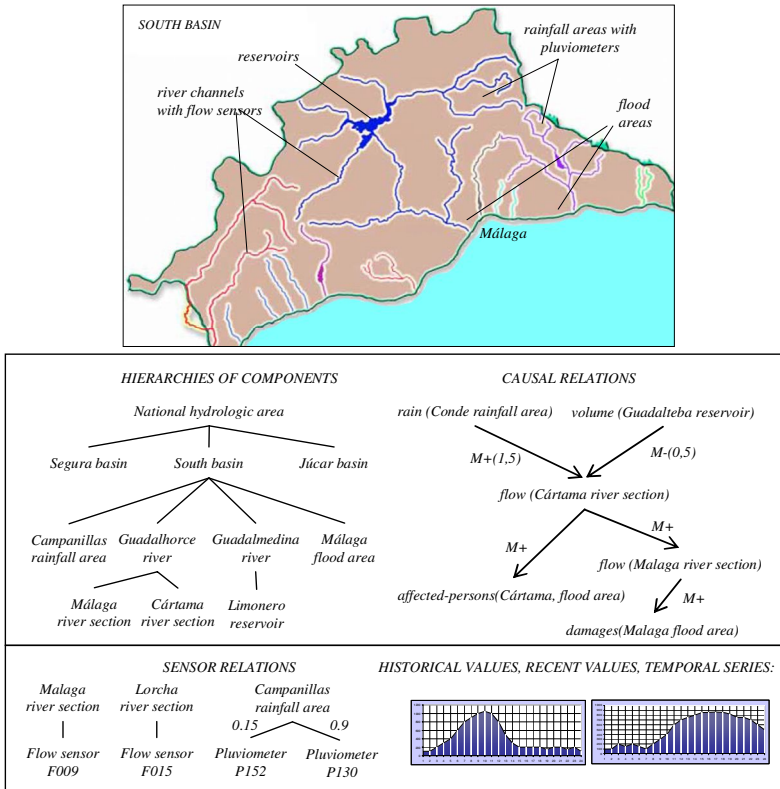


Fig. 1. Summary of the representation for the dynamic system in the domain of hydrology

2.2 The Interpretation Model

The interpretation model expresses how to determine the qualitative state of every node in the hierarchy of components. For the case of single components their state is determined directly by conditions about parameters. Normally, this is formulated with conditions about limit points that define the quantitative space corresponding to the state. This can be formulated by *qualitative interpretation rules*, i.e. sentences with the following format (x is a component and y_j are parameters):

$$\forall x, y_1, \dots, y_n (type(x, a) \wedge param(x, y_1) \wedge \dots \wedge param(x, y_n) \wedge COND_k(y_1, \dots, y_n) \rightarrow state(x, b))$$

where $type(x, a)$ means that the type of component x is a , $param(x, y)$ means that y is a parameter of x , $COND_k$ is a logical expression about the values of parameters y_1, \dots, y_n and $state(x, b)$ means that the state of the component x is b . An example in natural language is: *the state of a reservoir is near-limit-increasing if its volume is between 90% and 100% of its capacity and the time derivative of the volume is positive.*

For the case of complex components their state is determined by conditions about the state of simpler components. This can be formulated by *aggregation rules*, i.e. sentences based on the following format (x, y are components):

$$\forall x, y \ (type(x, a) \wedge member(y, x) \wedge type(y, b) \wedge state(y, c) \rightarrow state(x, d))$$

where $member(y, x)$ means that the component y is member of the component x . With this type of rules, a particular component could deduce different states based on the states of its different members. So these sentences must be interpreted following a particular control mechanism based on relevance as it is described in the following section. An example in natural language is: *the state of the basin is damages if there is a flood-area of the basin that presents the state of agricultural-losses*.

The interpretation model also is used to formulate how the value of a parameter x is computed based on the values of other parameters y_1, y_2, \dots, y_n (when x is not directly measured by sensors). This can be expressed with functional sentences where each sentence associates to a parameter x a function applied to the other parameters y_1, y_2, \dots, y_n . The function is taken from a library that includes arithmetic functions, statistical functions for both temporal and component abstraction, etc. An example of this sentence in natural language is: *the storage percent of a reservoir is the current volume of the reservoir multiplied by 100 and divided into the capacity of the reservoir*.

2.3 The Saliency Model

The saliency model represents a kind of control knowledge to determine when certain event is relevant to be reported to the operator. In general, we consider a relevant event as a significant deviation of the desired state established by the goals of the management strategy of the dynamic system. This definition is valid to report the relevant information about the behavior of the dynamic system during a long period of time. However, when operators monitor on real time the behavior of the system, we consider the notion of relevance as follows:

Definition. A *relevant event* is an event that (1) changes with respect to the immediate past and (2) produces a change (now or in the near future) in the distance between the state of the dynamic system and the desired state established by the management goals.

The implication of this definition is that, in order to evaluate the relevance of facts, it is necessary to predict the final effect of state transitions. However, based on our assumption for system modeling, we follow here a simplified and efficient approach with approximated knowledge for the system behavior. According to this, the representation of relevance establishes when a state can affect to the management goals, using a heuristic approach that summarizes sets of behaviors. This is formulated as logic implications that include (1) in the antecedent, circumstantial conditions about states of components and (2) in the consequent, the state of a component that should be considered relevant under such conditions. The general format is (x and y_j are components):

$$\forall x, y_1, \dots, y_n \ (type(x, a) \wedge REL_k(x, y_1, \dots, y_n) \wedge state(y_1, b_1) \wedge \dots \wedge state(y_n, b_n) \rightarrow relevant(state(x, c))$$

where $REL_k(x, y_1, \dots, y_n)$ relates a component x with other components y_1, \dots, y_n , according to physical properties (for instance a relation that represents the reservoirs that belong to a river). Thus, in hydrology, light rain is normally considered non relevant except, for example, if the weather forecast predicts heavy rain and the volume in a reservoir downstream is near the capacity.

Our notion of relevance gives also criteria to establish order among relevant events. This can be done with sentences that represent heuristic knowledge defining priority between two states based on their impact on the management goals. The representation uses conditional sentences that conclude about preference between states (represented by $A > B$, A is more relevant than B) with the following format (x and y are components):

$$\forall x, y (type(x, a) \wedge type(y, b) \wedge COND_k(x, y) \rightarrow state(x, a) > state(y, b))$$

where $COND_k$ is a logical expression (possibly empty) about the components x and y . For example, in hydrology, this allows to establish that heavy-rain at certain location x_1 is more relevant than the same rain at location x_2 . It also allows formulating a general priority scheme like: *damages > volume > flow > rain > weather-forecast*.

It is important to note that this priority scheme plays the role of control knowledge in the complete model. The aggregation rules of the interpretation model are used to determine the state of components based on the state of simpler ones. However, to avoid contradictory conclusions, these sentences need to be applied according to certain control mechanism. The relevance priority is used here for this purpose taking into account that sentences that interpret qualitative states with higher priority are applied first.

2.4 The General Inference

The general inference exploits the physical system properties (e.g., causal relations, member relations and changes in qualitative states) together with domain knowledge about relevance to produce the summary. In particular it performs a linear sequence of the following inference steps:

1. *Interpret*. For every single component its qualitative state is computed using as input the qualitative interpretation rules and the measures of sensors.
2. *Select*. Relevant states are selected. For every single component, the relevance of its state is determined by using the saliency model according to the following definition. A state $S(t_i)$ in the present time t_i of a component C is relevant if (1) the state $S(t_{i-1})$ of component C in the immediate past changes, i.e., $S(t_i) \neq S(t_{i-1})$, and (2) the predicate $state(C, S(t_i))$ is deduced as relevant according to the domain-dependent rules of the salience model. Let $R = \{S_1, S_2, \dots, S_n\}$ be the set of relevant states.
3. *Sort*. The set R of relevant states is sorted according to the domain-based heuristics of the salience model. More relevant states are located first in R .
4. *Filter*. Less relevant states that correspond to the same physical phenomenon are removed. For each state S_i in R (following the priority order in R) a second state S_k is removed from R if (1) S_k is less relevant than S_i (i.e., S_k is located after S_i in R), and (2) S_k is member of $causes(S_i)$ or S_k is member of $effects(S_i)$. Here, $causes(X)$

and $effects(X)$ are the sets that respectively contain all the (direct or indirect) causes and effects of X based on the causal relations of the system model.

5. *Condense*. The states of similar components are condensed by (1) *aggregation* and (2) *abstraction*. States of components with the same type are *aggregated* by the state of a more global component by using the aggregation rules of the interpretation model. Here, the salience model is used as control knowledge to select among candidate rules as it was described in the previous section. In addition to that, states of components of different type are *abstracted* by the most relevant state using the priority order in R . This produces what we call a *summarization tree*.

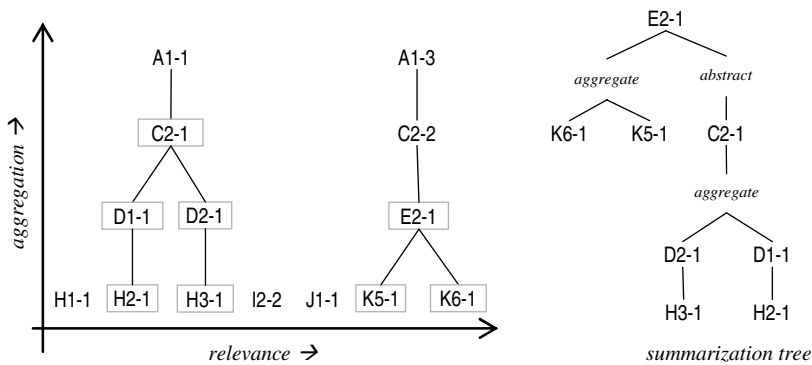


Fig. 2. Example of summarization tree

The example of figure 2 shows a summarization tree corresponding to a set of relevant states. In the example, the graphic at left hand side shows a partial search space. At the bottom, there are states of single components (K6-1 means the state 1 of component 6 of type K). The horizontal axis shows the relevance order (e.g, K6-1 is more relevant than K5-1). The squared states correspond to the elements of $R = \{K6-1, K5-1, H3-1, H2-1\}$. Upper nodes in these hierarchies are potential states inferred by aggregation rules. The corresponding summarization tree is presented at the right hand side. In this tree the most relevant and aggregated state is represented by the root E2-1.

3 Application in Hydrology

The previous general approach has been applied to the field of hydrology. In Spain, the SAIH National Programme (Spanish acronym for Automatic System Information in Hydrology) was initiated with the goal of installing sensor devices and telecommunications networks in the main river basins to get on real time in a control center the information about the hydrologic state. One of the main goals of this type of control centers is to help to react in the presence emergency situations as a consequence of river floods. The management goals in this case are oriented to operate reservoirs to avoid problems produced by floods and, if problems cannot be

avoided, to send information to the public institutions in order to plan defensive actions. Here, the generation of summaries of behavior is oriented to report relevant information of the river basin from the point of view of potential floods. This task can be considered as one of the components of a more complex intelligent system for emergency management [7].



Fig. 3. Example of summary in text mode



Fig. 4. Example of 3D animation on a virtual terrain presenting relevant information

In this context, information is received periodically at the control center about rainfall at certain locations, water levels and flow discharge in reservoirs and flows in certain river channels. A typical number of variables in a basin with a SAIH control center is about 500 measures every Δt (for example $\Delta t=30$ min). The analysis of a hydrologic situation requires usually data from the last 24 hours, so a typical amount

of data about 24,000 quantitative values. As a result of the summarizing process, relevant states are reported such as important rainfall at certain location, a significant increase of flow at certain location or, during the evolution of a particular storm, significant decrease of rainfall or flow at certain locations.

In order to present the summarized information, different modes have been considered such as text, 2D graphics and 3D animations on a virtual terrain. Figures 3 and 4 show examples of these types of presentations. To automatically construct the report, the computer system includes planner based on HTN (Hierarchical Task Networks) [8]. The planner follows a template-based strategy, with abstract presentation fragments corresponding to discourse patterns. According to the type of user the information is presented using different devices such as mobile phone (with *sms* messages), fax or a computer screen.

4 Summary and Discussion

In summary, the paper describes our approach for a summarization problem. The problem is summarizing the behavior of a complex dynamic system, where partial and approximate knowledge about structure and behavior is available. The main contributions of our work for this problem are: (1) a notion of relevance based on the distance to management goals which provides a particular strategy for summarization, and (2) the identification and representation of different types of available knowledge together with an inference procedure. The approach presented in this paper has been initially validated in the domain of hydrology with successful preliminary results with partial models. Currently we are working in a more extensive and complete evaluation of the solution and its integration with presentation methods.

Our approach is related to several general AI fields such as event summarization, model-based problem-solving methods and relevance reasoning. Within the field of event summarization [1] [9], there are techniques that go from domain dependent approaches (taking into account saliency and abstraction techniques) to domain independent solutions based on statistic analysis. Our approach is a domain dependent approach that follows specific inference strategies derived from the context of surveillance of dynamic systems.

On the other hand, in the field of model-based solutions, our approach is related to modeling approaches for qualitative physics [10] [11] [12] such as CML [3] and DME [4]. These general approaches are theoretical solid approaches that in practice usually need to be formulated with additional control mechanisms to avoid computational problems. Our approach is not oriented for prediction nor for diagnosis so it follows a simpler and more efficient representation for the behavior that requires less knowledge acquisition effort. Compared to methods for diagnosis [13] our approach does not look for hidden causes starting from given symptoms. Instead, it selects and summarizes the relevant information in the measured data.

Relevance reasoning has been studied from different perspectives such as philosophical studies or logic-based formal systems [14]. In artificial intelligence it has been considered in different problems such as probabilistic reasoning [15] or knowledge base reformulation for efficient inference [16]. Closer to our approach, relevance reasoning has been used in the representation of dynamic systems. For example, relevance reasoning is applied in compositional modeling (dynamic

selection of model fragments for simulation) [17] which is not the same task performed by our method. Our approach is closer to the case of explanation generators of device systems such as the system of Gruber and Gautier [18]. As in our approach, this system defines relevance based on state transitions. However, our method includes additional domain dependent mechanisms for relevance based on the management strategy, a filtering procedure based on causal knowledge, and additional abstraction techniques based on hierarchies of components.

Our approach is also related to techniques for summarizing time series data. For example, our work presents certain commonalities with the SumTime project [19]. Compared to our work, this project pays more attention to the natural language generation from temporal series while our work is more centered on using a particular representation of the dynamic system that provides adequate solutions for data interpretation, aggregation and filtering.

Other similar systems but restricted to the field of meteorology have been developed for summarizing [20] [21] [22]. For example, the RAREAS system is a domain dependent application that generates text summaries of weather forecast from formatted data. In contrast, our method has been conceived in general to be used in different domains such as road traffic networks, water-supply distribution networks, etc.

Acknowledgements. The development of this research work was supported by the the Ministry of Education and Science of Spain within the E-VIRTUAL project (REN2003-09021-C03-02). In addition to that, the Ministry of Environment of Spain (*Dirección General del Agua*) provided information support about the domain in hydrology. The authors wish to thank Sandra Lima for her valuable comments and her work on the implementation of the method.

References

1. Maybury, M. T.: "Generating Summaries from Event Data". Information Processing and Management: an International Journal. Volume 31. Issue 5. Special issue: Summarizing Text. Pages: 735 – 751. September 1995.
2. Schreiber G., Akkermans H., Anjewierden A., De Hoog R., Shadbolt N., Van de Velde W., Wielinga B.: "Knowledge engineering and management. The CommonKADS methodology" MIT Press, 2000.
3. Bobrow D., Falkenhainer B., Farquhar A., Fikes R., Forbus K.D., Gruber T.R., Iwasaki Y., and Kuipers B.J.: "A compositional modeling language". In Proceedings of the 10th International Workshop on Qualitative Reasoning about Physical Systems, pages 12-21, 1996.
4. Iwasaki Y. and Low C.: "Model Generation and Simulation of Device Behavior with Continuous and Discrete Changes". Intelligent Systems Engineering, Vol. 1 No.2. 1993
5. Gruber T. R. and Olsen G. R.: "An Ontology for Engineering Mathematics". In Jon Doyle, Piero Torasso, & Erik Sandewall, Eds., Fourth International Conference on Principles of Knowledge Representation and Reasoning, Gustav Stresemann Institut, Bonn, Germany, Morgan Kaufmann, 1994.
6. Borst P., Akkermans J. M., Pos A., Top J. L.: "The PhysSys ontology for physical systems". In R. Bredeweg, editor, Working Papers Ninth International Workshop on Qualitative Reasoning QR'95. Amsterdam, NL, May 16-19. 1995.

7. Molina M., Blasco G.: "A Multi-agent System for Emergency Decision Support". Proc. Fourth International Conference on Intelligent Data Engineering and Automated Learning, IDEAL 03. Lecture Notes in Computer Science. Springer. Hong Kong, 2003.
8. Ghallab M., Nau D., Traverso P.: "Automated Planning: Theory and Practice". Morgan Kaufmann, 2004.
9. Maybury, M. T.: "Automated Event Summarization Techniques". In B. Endres-Niggemeyer, J. Hobbs, and K. Sparck Jones editions, Workshop on Summarising Text for Intelligent Communication. Dagstuhl Seminar Report (9350). Dagstuhl, Germany. 1993.
10. Forbus K. D.: "Qualitative Process Theory". *Artificial Intelligence*, 24: 85-168. 1984.
11. de Kleer, J., Brown, J.: "A Qualitative Physics Based on Confluences". *Artificial Intelligence*. 24:7-83. 1984.
12. Kuipers B.: "Qualitative simulation". Robert A. Meyers, Editor-in-Chief, *Encyclopedia of Physical Science and Technology*, Third Edition, NY: Academic Press, pages 287-300. 2001.
13. Benjamins R.: "Problem-solving methods for diagnosis". PhD thesis, University of Amsterdam, Amsterdam, The Netherlands. 1993.
14. Avron A.: "Whither relevance logic?". *Journal of Philosophical Logic*, 21:243-281. 1992.
15. Darwiche, A.: "A logical notion of conditional independence". *Proceedings of the AAAI Fall Symposium on Relevance*, pp. 36-40, 1994.
16. Levy A., Fikes R., Sagiv, Y.: "Speeding up inferences using relevance reasoning: a formalism and algorithms". *Artificial Intelligence*, v.97 n.1-2, p.83-136, Dec. 1997
17. Levy A., Iwasaki Y., and Fikes R.: "Automated Model Selection Based on Relevance Reasoning", Technical Report, KSL-95-76, Knowledge Systems Laboratory, Stanford University. 1995.
18. Gruber, T. R., Gautier, P. O.: "Machine-generated Explanations of Engineering Models: A Compositional Modeling Approach". *Proceedings of the 13th. International Joint Conference on Artificial Intelligence*. 1993.
19. Sripada, S. G., Reiter, E., Hunter, J., and Yu, J., "Generating English Summaries of Time Series Data Using the Gricean Maxims", *Proceedings of the 9th International Conference on Knowledge Discovery and Data Mining SIGKDD*, Washington, D.C. USA, 2003.
20. Kittredge R., Polguere A., Goldberg E.: "Synthesizing weather forecasts from formatted data". In *Proc. International Conference COLING-86*, Bonn, August 1986.
21. Bourbeau L., Carcagno D., Goldberg E., Kittredge R., Polguere A.: "Synthesizing Weather Forecasts in an Operational Environment". In *Proc. International Conference COLING-90*, vol. 3, 318-320, Helsinki, August 1990.
22. Goldberg, E.; Driedger, N.; Kittredge, R.I.: Using natural language processing to produce weather forecast". *IEEE Intelligent Systems and Their Applications*. Volume 9, Issue 2, April 1994.

INFRAWBS Designer – A Graphical Tool for Designing Semantic Web Services

Gennady Agre

Institute of Information Technologies – Bulgarian Academy of Sciences
agre@iinf.bas.bg

Abstract. In order to make accessible new Semantic Web Services technologies to the end users, the level of tools supporting these technologies should be significantly raised. The paper presents the architecture of such a tool - an INFRAWBS Designer – a graphical ontology-driven tool for creating a semantic Web service description according to WSMO Framework. The tool is oriented to the end users – providers of Web services, who would like to convert their services into WSMO based semantic Web services. The most character features of the tool – intensive use of ontologies, automatic generation of logical description of a semantic service from graphical models and the use of similarity-based reasoning for finding similar service descriptions to be reused as initial templates for designing new services are discussed.

Keywords: Semantic Web Services, Web Service Modeling Ontology, Graphical Modeling, Case-based Reasoning.

1 INFRAWBS Project

INFRAWBS is a service oriented European research project, which primary objective is to develop an ICT framework enabling service providers to generate and establish open and extensible development platforms for Web Service applications [9]. Although INFRAWBS is conceived to design, maintain and execute Semantic Web Services (SWS) based on existing Web Services, most of its components have been designed to work as WSMO based SWS. WSMO is a European initiative for Semantic Web Services [13], which provides a formal ontology and language for describing the various aspects related to SWS.

Conceptually, the INFRAWBS Framework consists of coupled and linked INFRAWBS semantic Web units (SWU), whereby each unit provides tools and components for analyzing, designing and maintaining WSMO based SWS within the whole life cycle [9]. An INFRAWBS Semantic Web Unit provides users with:

- *Information structures* for effective discovering, storing and retrieving both semantic and non-semantic information needed for creating and maintaining SWS:
 - *Distributed SWS Repository (DSWS-R)* [10] is aimed at effective storing and retrieving all elements of the Semantic Web according to the WSMO Framework: Goals, Ontologies, SWS and Mediators. The repository also store some additional INFRAWBS specific data such as graphical models used for creating WSMO objects and “natural language” templates for WSMO Goals.

- *Similarity-based Organizational Memory (OM)* [12] contains non-logical representation of the WSMO objects stored in the DSWS-R and is used for their indexing and retrieval.
- *Semantic Information Router (SIR)* [14] is responsible for storing, categorization and retrieval of non-semantic Web services represented as WSDL files
- *Tools for creating and maintaining SWS and SWS Applications:*
 - *CBR based Service Designer (SWS-D)* [4] is aimed at designing a WSMO based SWS from an existing non-semantic Web service
 - *CBR based Service Composer (SWS-C)* [4] is aimed at creating a SWS through composition of existing WSMO-based SWS
 - *CBR based Goal Editor* is aimed at creating predefined WSMO-based goals and their “natural language” templates needed for designing SWS-based applications
 - *CBR based Recommender tool* [3] is a similarity-based tool facilitating operation of all INFRAWEBS “semantic-based” tools by utilizing “past experience”. Non-semantic data stored in OM is used as the problem description for determining the most similar solution (SWS or its graphical model) to the current problem.
- *Problem-Solving Methods* used for creating and maintaining Semantic Web Services:
 - Logic-based discovery
 - Application-specific decision-support methods used for service composition, compensation, monitoring etc.
 - Ontology keywords based discovery [6]
 - Several methods for calculating similarity and/or assignments – structural, linguistic, statistic etc.

SWU is embedded in the *INFRAWEBS Environment* [16] which provides means for communicating with different kinds of INFRAWEBS users and other INFRAWEBS Semantic Web Units as well as for executing SWS with ensuring security and privacy of these operations. The INFRAWEBS architecture reflects a novel approach for solving problems occurring during creating SWS application - the tight integration of similarity-based and logic-based reasoning. The similarity-based reasoning is used for fast finding an approximate solution, which is farther clarified by the logic-based reasoning.

The present paper is aimed at presenting the architecture, basic design principles and some implementation details of the INFRAWEBS Designer – a graphical ontology-driven tool for creating WSML-based logical description of SWS according to WSMO Framework. This tool is oriented to the end users – providers of Web services, who would like to convert their services into WSMO based semantic services. The structure of the paper is as follows: in the next section the main design principles and the conceptual architecture of the INFRAWEBS Designer are presented. Then we discuss the most characteristic features of our tool – the intensive usage of ontologies, the graphical way for creating service description and the reuse of semantic descriptions of existing services for facilitating the process of designing new semantic service description. In the conclusion we present some implementation details and compare our tool with similar products.

2 The Conceptual Architecture and Main Design Principles

According to the WSMO Framework, a semantic service description consists of three main parts [13]:

- *Top level concepts* – describing service name spaces, used ontologies and mediators as well as service nonfunctional properties.
- *Service capability* – describing what the service can do. The description is separated into four sections – assumptions, preconditions, postconditions and effects represented as WSMO logical expressions (axioms). The connections between them are the common set of ontologies used as well as a set of common variables (optional) called shared variables.
- *Service choreography* – describing how the user can communicate with the semantic service. WSMO choreography is based on Abstract State Machine methodology [15] and consists of state signature and state transition rules. The State signature defines the state ontology used by the service together with the definition of the types of modes the concepts and relations may have, while the Transition rules that express changes of states by changing the set of instances.

The main parts of a SWS description (service capability and transition rules) are represented via complex logical expressions written on WSMO language, which combines features of F-logic, Description logic and Logic Programming [5]. Thus “direct” creating WSMO description of SWS (e.g. by means of a text editor) is not an easy task and requests strong knowledge in formal logic, which significantly restricts a circle of people able to accomplish such an activity. That is why we have decided to avoid this problem by constructing a special tool for designing SWS – the INFRAWEBS Designer, which main design principles are:

- *User-friendliness*: it is assumed that the users of our tool will be semantic Web service providers, who will not be specialists in first-order logic, so we propose a graphical way for constructing and editing the service description abstracting away as much as possible from a concrete syntax of logical language used for implementing it.
- *Intensive use of ontologies*: our analysis has shown that the main difficulties of the process of constructing complex logical expressions (axioms) are associated with use of correct names of concepts, attributes, relations and parameters as well as their types rather than with expressing logic itself. That is why the process of constructing the logical description of SWS in INFRAWEBS Designer is *ontology-driven*, which means that in each step of this process the user may select only such elements of existing ontologies that are consistent with already constructed part of the description.
- *Reusability*: creating a SWS description is a complex and time-consuming process, which can be facilitated by providing the service designer with an opportunity to reuse the existing descriptions of services (or their parts) created by the designer himself or by other users. The INFRAWEBS Designer provides the user with such an opportunity by applying the case-based reasoning approach.

The conceptual architecture of the INFRAWEBS Designer, which implements the mentioned above principles, is shown at the Fig. 1.

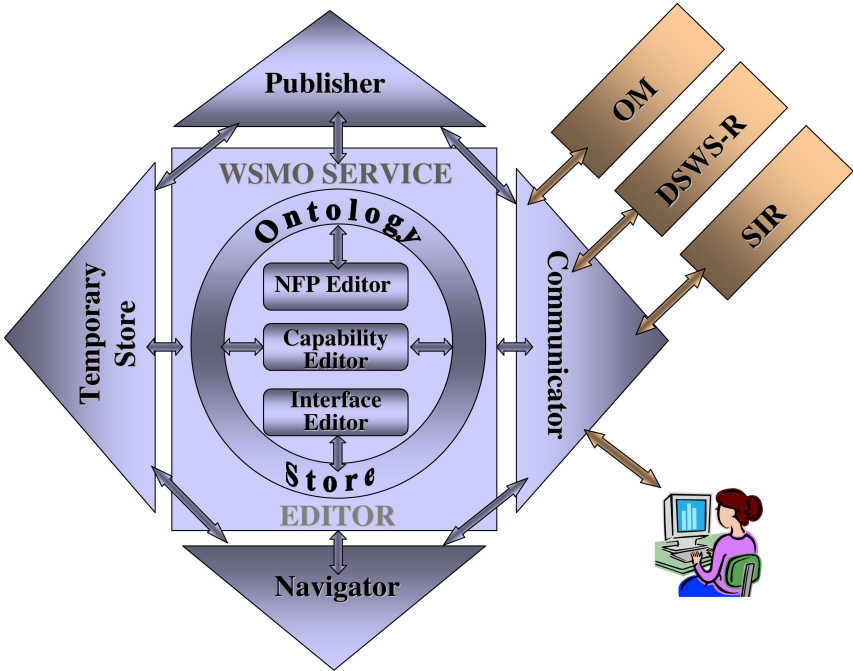


Fig. 1. Conceptual Architecture of the INFRAWEBS Designer

Temporary Store is an abstract module representing a place where all local files produced or used by the INFRAWEBS Designer are stored. Such files include a current WSML description of a semantic service under design, WSML descriptions of all ontologies used for the service description, files with graphical models of WSML logical expressions used for representing the service capability and choreography etc..

Navigator is an abstract module centralizing an access to other modules of the INFRAWEBS Designer. It stores to and retrieves from the Temporary Store all locally stored files and switches the control between the Communicator and different INFRAWEBS specialized editors. The Navigator is also responsible for creating a so called “Service Tree”, which may be seen as an initial skeleton of a newly created semantic service description. The whole process of service design in the INFRAWEBS Designer is organized as “populating” this tree.

Communicator is responsible for communication with the user and such external for the Designer INFRAWEBS components as OM (Organizational Memory), DSWS-R (Distributed Semantic Web Service Repository) and SIR (Semantic Information Router):

- OM is the INFRAWEBs case-based memory implemented as a Web service. The Designer uses the OM for finding semantic service descriptions similar to the service under construction as well as for WSML ontologies needed for realization of such a process. The graphical model of a “similar” service (or different parts of it) is used as a template for fast creating the description of a new semantic service. The Communicator creates and sends to the OM different criteria for searching the required object description and receives back a list of identifiers of existing semantic objects (services or ontologies), which have the best match against the specified criteria.
- DSWS-R is a storage for WSML descriptions of WSMO based semantic objects (services, ontologies, goals and mediators), which are indexed by the OM. The Communicator uses the DSWS-R for retrieving descriptions of semantic services and ontologies given the service or ontology identifier provided by the OM.
- SIR is an INFRAWEBs component responsible for categorization, annotation and retrieval of non-semantic (WSDL) descriptions of Web services. The Communicator sends to SIR the criteria for searching the desired Web service and received the Web service description, which is further used by the Interface Editor for creating the semantic service choreography.

Publisher is a module responsible for storing a semantic service description in an external (for the Designer) storage – the INFRAWEBs DSWS-R. In order to provide the correctness and completeness of such a description the Publisher validates different parts of the description and finalizes it with some information needed to guarantee the proper use of this service in the future.

WSMO Service Editor is an abstract module responsible for creating and editing WSML description of a semantic Web service according to the WSMO Framework. The Editor combines three specialized graphical ontology-driven editors for creating and editing different parts of such a description as well as a specialized module for in-memory loading, visualization and manipulation of WSML ontologies, which are used by these editors for their operation:

- *NFP Editor* is responsible for graphical creating and editing the WSMO-based description of non-functional properties of a semantic Web service.
- *Capability Editor* is responsible for graphical creating and editing the capability description of a semantic Web service. The Editor is an extension of the INFRAWEBs Axiom Editor [2], which allows graphical creating, editing and reusing of complex WSML-based logical expressions.
- *Choreography Editor* is responsible for graphical creating and editing the description of the choreography of a semantic Web service, which consists of an ontology-based state signature and state transition rules.
- *Ontology Store* is responsible for loading, visualizing and using WSML ontologies. It provides the basic graphical elements corresponding to ontology concepts, instances and relations for creating graphical models of a semantic service description, supports automatic on-demand loading of the required ontologies as well as actualization of the ontology concepts’ structure during the process of the graphical model creation.

3 The Use of Ontologies

A process for converting a non-semantic Web service into a WSMO-based semantic Web service may be seen as an interactive ontology-driven process of service annotation. Creation of such an annotation is crucially depends on the availability of proper ontologies. In practice finding the appropriate set of ontologies is one of the initial steps the user has to do before creating a semantic description of a Web service.

The INFRAWEBs Framework (IIF) assumes that all ontologies are stored in the DSWS-R. Thus, the process of finding ontologies is implemented in the Designer as a complex procedure for communicating with the IIF, in which the user describes the expected content of the required ontology, the Designer (or more precisely, the Communicator module of the Designer) translates this description into a query and sends it to the INFRAWEBs OM component, which plays a role of an indexer of the DSWS-R in the IIF. The OM matches the query against its internal representation of ontologies stored in DSWS-R (cases) and returns a set of ontology identifiers, which are the most similar to the query. After the user has inspected and selected the desired set of ontologies, they are loaded to the Designer's Ontology in-memory Store from the DSWS-R.

Ontologies describe inheritance between concepts. A concept usually has one or more super-concepts that can be defined in other "imported" ontologies mentioned in the corresponding section of the ontology description. Normally the imported ontologies are not loaded into the Ontology Store and as a result, all concepts having their super-concepts defined in such ontologies can not inherit any of the super-concepts' attributes. In order to avoid this deficiency the INFRAWEBs Designer provides a mechanism for *on-demand loading* imported ontologies when a concept, which super-concepts are defined in the imported ontologies, is selected.

All ontologies are loaded into the Ontology Store, which is a *global structure* accessible for all semantic services loaded into the Designer. Such an organization allows to design in parallel several new semantic services using the same set of ontologies from the Ontology Store. It is very convenient in cases, when the user is going to design several semantic services from a single complex Web service (e.g. Amazon.com), using different sets of the service operations.

Ontologies are visualized as trees, which nodes represent ontology concepts, relations and instances. In the tree-structured visualization every child element appears as many times in the tree as there are concepts in its *SuperConcepts* property. A visualized ontology may be browsed showing all properties associated with each ontology element in a special window. Usually a semantic service description is constructed by elements (concepts, relations etc.) from several ontologies as well as by extensive usage of built-in WSMO data types and predicates. To unify the access to concepts, relations and WSMO built-in constructs, the latter are treated as regular concepts and relations. Two such "built-in" ontologies are automatically loaded upon startup allowing the user to use all basic data types (strings, numeric values etc.) as ordinary ontology concepts and all available built-in predicates (such as numeric comparisons) - as ordinary ontology relations.

The most important elements of an ontology tree are concepts, instances and relations since they are the building elements of a semantic service description. All of

them can be drag-and-dropped from the Ontology Tree into the working graphical area of the corresponding graphical editor thus creating graphical model elements.

The INFRAWEBS Designer does not provide any means for creating and editing WSMO ontologies¹. All ontologies loaded into the Ontology Store are read-only, thus no ontology element can be changed. An exception is a case, when a list of attributes of an ontology concept is *automatically expanded* to include all attributes inherited from a concept's super-concept. Such an effect is a result of the on-demand loading the imported ontology in which this super-concept is defined.

4 Graphical Creation of a Semantic Service Description

4.1 Service Tree

As it has been already mentioned a description of a WSMO based semantic service contains three types of information related to service identification (service nonfunctional properties, name spaces etc.), service advertisement (a logical description of service capability that is used for semantic service discovery), and a service choreography (a logical description of service behavior that is used for communication with and execution of the service. Although such information is very heterogeneous it is displayed in the Designer in a uniform way – as a tree in which internal (functional) nodes represent the roles of each specific portion of the information and are used for editing the tree structure, while the tree leaves serve as pointers to the content of each information portion (Fig. 2). For example, a service precondition is represented by five functional nodes - “Shared variables”, “Preconditions”, “Assumptions”, “Post-conditions” and “Effects”. The last four nodes can have leaves which are the named pointers to the corresponding WSMO logical expressions (axioms) represented by their graphical models. The right-click menus associated with these tree nodes allow the user to create, remove or edit the axioms with the specified role in the service description.

The shared variables node of the service tree can contain sub-trees, which leaves contain information on a variable name and a place, where the variable used (axiom or axiom definition if the axiom has more than one definition). The right-click menu associated with this node allows only deleting or renaming selected variables since

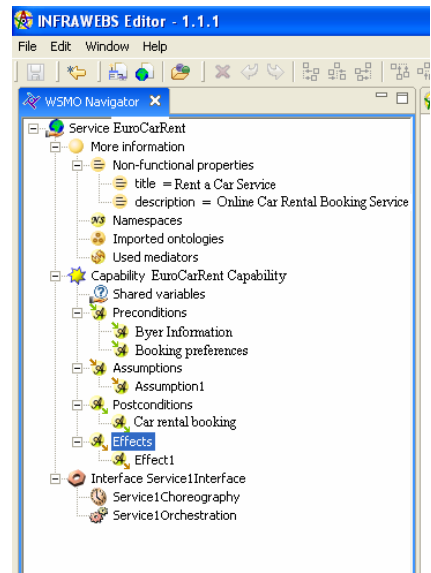


Fig. 2. An example of the service tree

¹ However it can be done by means of WSMO Studio (<http://www.wsmstudio.org>) integrated with the Designer.

creating a shared variable can be done only from a graphical model of an axiom by means of a special operation.

4.2 Graphical Models of WSMML Logical Expressions

In the description of a WSMO based semantic service logical expressions are used for representing service capability as well as conditional and conclusion parts of transition rules describing the service choreography. The conceptual syntax for WSMML has a frame-like style [5]. The information about a class and its attributes, a relation and its parameters and an instance and its attribute values is specified in a large syntactic construct, instead of being divided into a number of atomic chunks. WSMML allows using of variables that may occur in place of concepts, attributes, instances, relation arguments or attribute values. Although the machines can easily handle such axioms, creating and comprehending the axioms are very difficult for humans. That is why we have developed an approach in which the text (WSMML) representation of such expressions is automatically generated from their graphical models. The approach has been implemented as a special software component called Axiom Editor. A detailed description of the functionality of this component as a stand-alone software component is presented in [2]. It was also included as an Eclipse third party plug-in in the last version of WSMO Studio². The present section contains brief description of main ideas of the approach as well as a description of an additional functionality of this component caused by its usage as a main tool for graphical creating semantic service capability and choreography description in the INFRAWEBS Designer.

4.2.1 Representation of WSMML Logical Expressions

A WSMML logical expression is graphically modeled as a direct acyclic graph (DAG), which can contain four types of nodes:

- A single node called *Root*, which may have only outgoing arcs. The node corresponds to WSMML statement **defineBy**. Graphically the root node is represented as a circle named “Start”.
- Intermediate nodes called *variables*. Such nodes can have several incoming arcs and outgoing arcs. Each variable has a unique name and poses a frame-like structure consisting of slots represented by attribute–value pairs. Such a variable corresponds to a notion of compound molecule in WSMML [5] consisting of an a-molecule of type Var_i **memberOf** Γ and conjunction of b-molecules of type $Var_i[p_1 \text{ hasValue } Var_{j1}]$ and $Var_i[p_k \text{ hasValue } Var_{kl}]$ respectively, where $Var_i, Var_{j1}, Var_{kl}$ are WSMML variables and Γ is a concept from a given WSMML ontology. Graphically each variable is represented as a rectangle with a header containing variable name and type (i.e. the name of concept, which has been used for creating the variable), and a row of named slots.
- Intermediate nodes called *relations*. A relation node corresponds to a WSMML statement $r(Par_1, \dots, Par_n)$, where r is a relation from a given ontology, and Par_1, \dots, Par_n are WSMML variables – relation parameters. Graphically each relation node is represented as a rectangle with a header containing relation name and a row of relation parameters.

² The latest release can be downloaded from <http://www.wsmstudio.org/download.html>

- Intermediate nodes called *operators* that correspond to WSMML logical operators *AND*, *OR*, *IF-THEN*³ and *NOT*. Each node can have only one incoming arcs and one (for *NOT*), two (for *IF-THEN*) or more (for *AND* and *OR*) outgoing arcs. Graphically each operator is represented as an oval, containing the name of the corresponding operation.
- Terminal nodes (leaves) called *instances* that can not have any outgoing arcs. An instance corresponds to the WSMML statement *Var hasValue Instance*, where *Var* is a WSMML variable and *Instance* is an instance of a concept from a given ontology. Graphically an instance is represented by a rectangle with header containing the instance name and type.

Directed arcs of a graph are called *connections*. A connection outgoing from a variable or relation has the meaning of refining the variable (or relation parameter) value and corresponds to WSMML logical operator *AND*. A connection outgoing from an operator has the meaning of a pointer to the operator operand. The proposed model allows considering the process of axiom creation as a formal process of DAG expanding (and editing) and to formulate formal rules for checking syntactic and semantic (in relation to given ontologies) correctness of the constructed logical expressions.

4.2.2 A Model for Constructing Logical Expressions

Constructing a logical expression is considered as a repetitive process consisting of combination of three main logical steps – definition, refinement and logical development. The *definition* step is used for defining some general concepts needed for describing the meaning of axioms. During this step the nature of a main variable defining the axiom is specified. Such a step is equivalent to creating a WSMML statement *?Concept memberOf Concept*, which means that the WSMML variable *?Concept* copying the structure of the *Concept* from a given WSMML ontology is created. Attributes of the concept, which are “inherited” by the axiom model variable, are named *variable attributes*. By default the values of such attributes are set to free WSMML variables with type defined by the definition of such attributes in the corresponding ontology.

The *refinement* step is used for more concrete specification of the desired properties of such concepts and may be seen as a specialization of too general concepts introduced earlier. This step is implemented as a recursive procedure of refining values of some attributes (or relation parameters) defined in previous step(s). In terms of our model each cycle in such a step means an expansion of an existing non-terminal node – variable (or relation). More precisely that means a selection of an attribute of an existing model variable, and binding its value (which in this moment is a free WSMML variable) to another (new or existing) node of the axiom model. The main problem is to ensure semantic correctness of the resulted (extended) logical expression. Such correctness is achieved by applying a set of context-sensitive rules determining permitted expansion of a given node.

The *logical development* step consists of elaborating logical structure of the axioms, which is achieved by combination of general concepts by means of logical operators *AND*, *OR*, *IF-THEN* and *NOT*. Such operators may be added to connect two independently constructed logical expressions or be inserted directly into already

³ This operator corresponds to *IMPLIES* and *IMPLYBY* operators in WSMML.

constructed expressions. The operation is controlled by context-dependent semantic and syntactic checks that analyze the whole context of the axiom.

It should be underlined that during this step the user is constructing the axiom by logical combination of main axiom objects defined in the previous steps. In other words, the logical operators are used not for refining or clarifying the meaning of some parameters of already defined objects, but for complicating the axiom by specifying the logical connections between some axiom parts which are independent in their meaning. An example of a graphical model is shown on Fig.3.

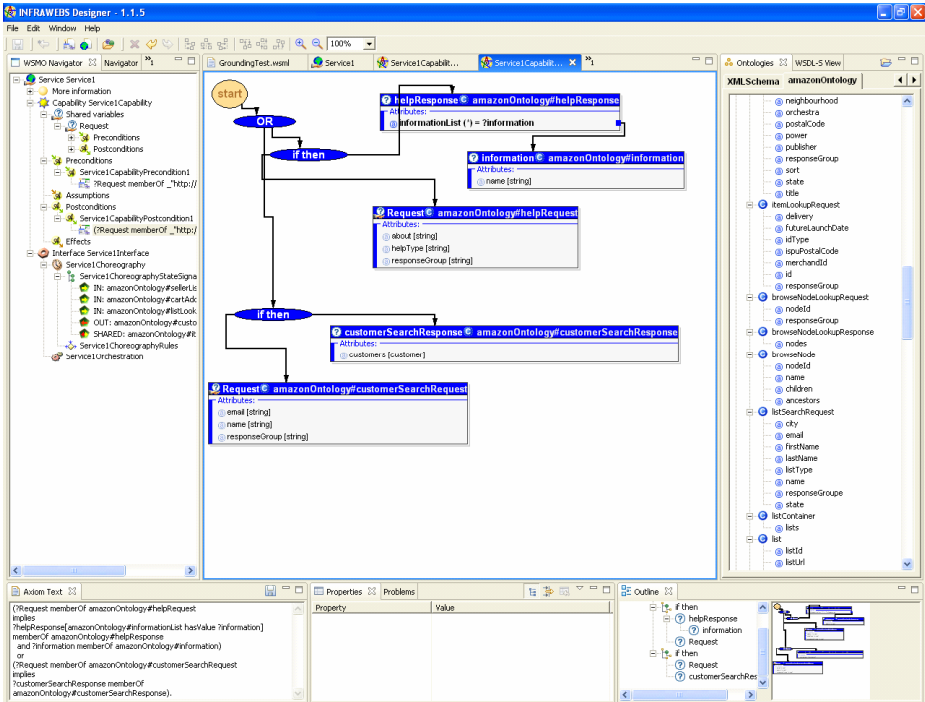


Fig. 3. An example of a graphical model created by the INFRAWEBS Designer


5 Reusing Descriptions of Existing Semantic Services

Although the usage of graphical models makes easier the process of designing WSMML description of a semantic service, it still remains a rather complex and time-consuming activity. The next step towards facilitating this process is to provide the service designer with an opportunity to reuse the description of the existing semantic services. More precisely, the idea is to provide the user with graphical models of service description parts (e.g. capability axioms or transition rules) *similar* to what she wants to construct, which can be further adapted by graphical means provided by the INFRAWEBS Designer.

In order to be reused, WSMO descriptions of semantic Web services and other objects (goals, ontologies, mediators) are considered in the INFRAWEBS Framework

not only as logical (WSML) representations of these objects but also as a *special class of text documents*, containing natural language and ontology-based words. Such special “text” representation is extracted by OM from each object stored in the DSWS-R and serves as a basis for constructing case representation of such an object. An INFRAWEBBS case is a triple $\{T, P, S\}$, where T is a type of the WSMO object stored (service, goal or ontology), which determines the structure of object representation P ; P is a special representation of a WSMO object as a *structured text*, and S is a pointer to the place in a local DSWS-R where the WSML (and graphical) description of the object is stored⁴.

The user describes a semantic service to be found by filling a standard request form (see Fig.4), which is sent to the OM playing a role of a case-based memory in the INFRAWEBBS Framework. The form consists of three sections allowing constructing different queries based on the amount of information the user has in the current moment.

The first section (“Text”) allows the user to describe the desired functionality of a service to be found by means of natural language keywords. All non-functional properties (of type “string”) occurred in the WSML descriptions of semantic services stored in the DSWS-R will be matched against these keywords, and services with the best match will be returned. The second section (“Ontologies”) allows the user to find services using a set of ontologies similar to that specified in the request form. Since in the INFRAWEBBS Designer the user can work with several services in parallel, the filling of this section is done by pressing the corresponding button  when the service containing the desired set of ontologies is active. By switching among descriptions of services loaded into the Designer, the user can construct the desired set of ontologies, which combines ontologies from different services. During processing the request form the Communicator analyses each of the specified services and extracts from their descriptions a list of names of ontologies imported by each service. These names are used by the OM as keywords in the process of finding the existing semantic services using the same (or similar) set of ontologies.

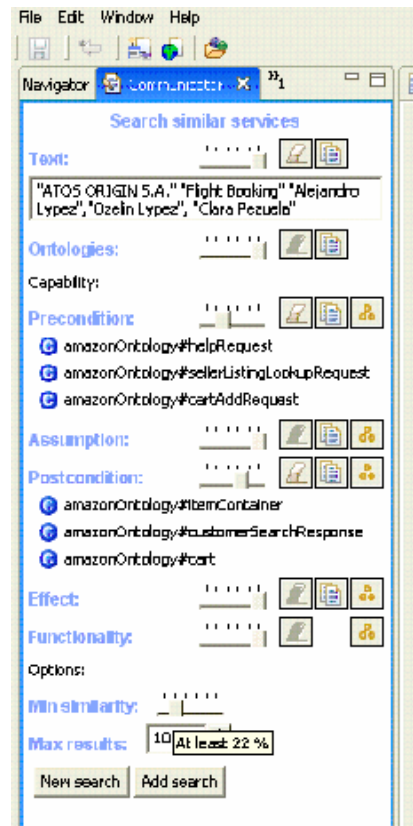



Fig. 4. A conceptual representation of a service request form


⁴ More detailed and formal definitions of the INFRAWEBBS case as well as of similarity functions can be found in [1]. The details on organization of the INFRAWEBBS case-based memory are presented in [3].


The third section (“Capability”) is devoted to ontological description of the capability of the desired service. It is split on the five subsections – the first four of them correspond to the sections of the service capability description according to the WSMO framework, and the last one (“Functionality”) allows constructing a general description of the capability of a service to be found.

The first four sub-sections can be filled by the following two ways (which can not be combined):



- *Semiautomatic* - by selecting the ontological elements (concepts, instances or relations) from the ontologies shown in the Ontology Store (the  button). In such a way the user can specify that she expects that the desired service should contain similar set of ontology keywords in the WSML (logical) description of the corresponding section of its capability.
- *Fully automatic* – by specifying the name of a service, whose corresponding capability description section will be used as an *example* of what the desired service should have in its capability section description.

When the user has no clear idea about the content of a concrete section of the desired service capability description or no “example” service descriptions exist, she can express her *general idea* of what ontology keywords should the desired service capability description have as *a whole*. Such general description is the content of the subsection “Functionality”. The subsection can be filled by selecting ontological elements from all available ontologies shown in the Ontology Store.

The user can edit the content of each section or delete all its data by pressing the “delete” button .

The filled request form is translated into a complex XML query, in which each form subsection determines a separate (local) criterion for evaluating the similarity. The overall aggregated similarity is calculated based on a weighted sum of all local criteria [1]. The user can set the weight (from 0 to 1) for each criterion by means of a slider  placed on the right from the corresponding form subsection (or section).

At the bottom of the form there is an additional section “Options”, which allows the user to adjust two global parameters of the search. The first one, implemented as a slider, determines the minimum similarity threshold, which should be exceeded by the service in order to be returned as a search result. The second one sets a maximum number of elements that may be returned.

The query results returned by the OM are represented as an annotated list of services IRI ordered by the value of their ontological similarity coefficient or by the value of the lexical similarity coefficient if only the text section of the query form has been filled by the user. The Communicator allows the user to send several queries to the OM without losing the results of previous query implementing in such a way the effects of searching alternatives. When the  button is pressed the results of the previous search are merged with the results of a new search. And again all results are ordered according to the values of the similarity coefficients. When the  button is selected, the previous query results are deleted and replaced by the results of a new query.

6 Related Works and Conclusion

The necessity of intensive development of tools supporting semantic Web service technology and specially WSMO initiative has been clearly recognized in WSMO community. The Ontology Editing and Browsing Tool⁵ is a plug-in for the Eclipse framework currently under development by the Ontology Management Working Group (OMWG). The tool can be used to edit Ontologies described within WSML documents.

A group of researches from DERI is developing the Web Services Modeling Toolkit (WSMT)⁶ - a framework for the rapid creation and deployment of homogeneous tools for Semantic Web Services. Besides several plug-ins for creating, visualizing and mediating ontologies, an initial set of tools for the WSMT includes a WSML Editor [7], which aims to provide a useful tool for describing Semantic Web Services using the WSMO ontology and publishing these descriptions to a WSMO repository and also to facilitate user experimentation with the WSMO ontology and WSML language. The main approach for realizing these aims is by using structural text editors, which of course simplify the process for creating WSML description of a service, but still request strong knowledge of WSML.

Selecting WSMO as a basic methodology for describing SWS in INFRAWEBS Framework has led to necessity to develop some basic tools providing full compatibility of INFRAWEBS Platform with existing WSMO environment. For these purposes it has been decided to adopt WSMO4J⁷ and WSMO Studio [17]. WSMO4J provides API and an implementation for WSMO data representation and modeling, as well as a framework allowing for parsing and serialization into a variety of formats and has no alternative in this role. WSMO Studio is a leading service environment, among the very few ones publicly available. It is well aligned with a number of other components and architectures (such as, the WSMX execution environment and many others developed in DIP⁸. WSMO Studio is based on WSMO4J and Eclipse⁹; which is the preferred GUI implementation platform for INFRAWEBS Designer as well. The INFRAWEBS Designer is implemented in J2SDK 1.4.2 runtime environment and uses basic platform components, plug-in infrastructure, graphical user interface components (menus, buttons, tree views, event handling) from Eclipse RCP (Rich Client Platform). For development of visual designers the Eclipse GEF (Graphical Environment Framework) is used.

Each of main INFRAWEBS Designer Editors – NFP Editor, Capability Editor and Choreography Editor is implemented as a WSMO Studio plug-in, which allows the end user to use other third party plug-ins e.g. for creating WSMO ontologies and mediator. The same is true for the INFRAWEBS Axiom Editor, which allows using it for creating and editing axioms included in WSML ontologies¹⁰. The choreography

⁵ <http://wwwhttp://sourceforge.net/project/g.org>

⁶ <http://sourceforge.net/project/>

⁷ <http://wsmo4j.sourceforge.net>

⁸ <http://dip.semanticweb.org>

⁹ <http://www.eclipse.org/>

¹⁰ Of course, it can be done using a restricted set of WSML logical operations supported by the Axiom Editor.

description created by the INFRAWEBS Designer conforms the latest WSMO choreography specifications [15].

The first prototype of the Designer with full functionality will be ready until July 2006 and will be applied for developing the Frequent Flyer Program [6] in which the customers can create and reuse travel packages. The application is built upon a Service Oriented Architecture, accessing, discovering, composing and invoking Semantic Web Services for the management of the Travel Packages.

Acknowledgements. This work is supported by the EC funded IST project INFRAWEBS (IST FP6-511723) within the framework FP 6.

References

1. G. Agre. Using Case-based Reasoning for Creating Semantic Web Services: an INFRAWEBS Approach. In: *Proc. of EUROMEDIA '2006*, April 17-19, 2006, Athens, Greece, 130-137.
2. G. Agre, P. Kormushev, I. Dilov. "INFRAWEBS Axiom Editor - A Graphical Ontology-Driven Tool for Creating Complex Logical Expressions. *International Journal "Information Theories and Applications"* Vol. 13, No. 2. ISSN 1310-0513 (2006), pp. 169-178.
3. G. Andonova, G. Agre, H.-j. Nern, A. Boyanov. Fuzzy Concept Set Based Organizational Memory as a Quasi Non-Semantic Component within the INFRAWEBS Framework. In: *Proc. of the 11th IPMU International Conference*, Paris, France, July 2-7, 2006 (in print).
4. T. Atanasova, G. Agre, H.-J. Nern. INFRAWEBS Semantic Web Unit for Design and Composition of Semantic Web Services. In: *Proc. of EUROMEDIA 2005*, Toulouse, France, pp. 216-220. April 11-13, 2005.
5. Bruijn, J.; Lausen, H.; Krummenacher, R.; Polleres, A.; Predoiu, L.; Kifer, M.; Fensel, D. 2005. D16.1 – *The Web Services Modeling Language (WSML)*. WSML Draft.
6. C. Fülöp, L. Kovács, A. Micsik. "The SUA-Architecture Within the Semantic Web Service Discovery And Selection Cycle. In: *Proc. of EUROMEDIA 2005*, Toulouse, France., April 11-13, 2005.
7. M. Kerrigan. Developers Tool Working Group Status. Version 1, Revision 4. http://wiki.wsmx.org/index.php?title=Developer_Tools.
8. J.-M. López-Cobo, A. López-Pérez, J. Scicluna. A semantic choreography-driven Frequent Flyer Program. In: *Proc. of FRCSS 2006 -1st International EASST-EU Workshop on Future Research Challenges for Software and Services*, April 1st 2006, Vienna (Austria).
9. A. López Pérez, J.-M. López-Cobo, Y. Gorronogoitia. A Framework for Building Semantic Web Services Applications. In: *Proc. of the 2006 International Conference on Semantic Web & Web Services*, Monte Carlo Resort, Las Vegas, Nevada, USA , June 26-29, 2006, (in print).
10. Z. Marinova, D. Ognyanoff and A. Kiryakov. *INFRAWEBS Deliverable D4.1-2.1. Specification of DSWS-R and reduced Registry*, August 2005.
11. H.-J. Nern, G. Agre, T. Atanasova, A. Micsik, L. Kovacs, T. Westkaemper, J. Saarela, Z. Marinova, A. Kyriakov. Intelligent Framework for Generating Open (Adaptable) Development Platforms for Web-Service Enabled Applications Using Semantic Web Technologies, Distributed Decision Support Units and Multi-Agent-Systems. *W3C Workshop on Frameworks for Semantics in Web Services*, Digital Enterprise Research Institute (DERI), Innsbruck, Austria, pp. 161-168. June 9-10, 2005.

12. H.-J. Nern, G. Jesdinsky, A. Boyanov. *INFRAWEBS Deliverable D2.1.1 Specification and Realised Reduced OM*. November 2005.
13. D. Roman, U. Keller, H. Lausen (eds.) *Web Service Modeling Ontology*, 2005,.WSMO Final Draft.
14. J. Saarela, S. Saarela and T. Westkämper. *INFRAWEBS Deliverable D3.1.1. Specification and Specific Application of the Reduced SIR*. August 2005.
15. D. Roman and J. Scicluna (eds). *D14v0.3. Ontology-based Choreography of WSMO Services*. WSMO Final Draft 19th May 2006 http://www.wsmo.org/TR/d14/v0.3/d14v03_20060519.pdf
16. J. Scicluna, T. Haselwanter and A. Polleres. *INFRAWEBS Deliverable D7.1-2.1. Reduced Rule Base, QoS Metrics, Running SWS-E and QoS Broker*. August 2005.
17. <http://www.wsmostudio.org>

Author Index

- Adam, Carole 24
Agre, Gennady 275
Amgoud, Leila 13

Balbo, Flavien 3
Batista, Fernando 213
Belabbès, Sihem 13
Ben Hariz, Sarra 162
Benhamou, Belaïd 33

Castro, Carlos 45, 56
Ciesielski, Krzysztof 245
Ciravegna, Fabio 2
Crawford, Broderick 45, 56
Czerski, Dariusz 245

d'Aquin, Mathieu 190
Dramiński, Michał 245
Dzbor, Martin 66

Elouedi, Zied 162

Flores, Victor 265
Fraser, Colin 150
Fusaoka, Akira 255

Gaudou, Benoit 24

Halpin, Harry 150
Hearst, Marti 233
Herzig, Andreas 24
Hiratsuka, Satoshi 255
Houvardas, John 77

Jauregi, Ekaitz 118

Kebair, Fahem 98
Kim, In-Cheol 181
Kłopotek, Mieczysław A. 245

Laborie, Sébastien 128
Lazkano, Elena 118
Lieber, Jean 190
Longin, Dominique 24
Lu, Hsin-Hung 255

Mamede, Nuno J. 213
Martínez-Otzeta, José María 118
Mellouli, Khaled 162
Molina, Martin 265
Monfroy, Eric 45, 56
Motta, Enrico 1, 66

Nakov, Preslav 233
Napoli, Amedeo 190, 201
Nauer, Emmanuel 201

Paris, Lionel 33
Paulo Pardal, Joana 213
Pietquin, Olivier 172
Pinto, H. Sofia 213
Popov, Borislav 222
Prade, Henri 13

Ribeiro, Ricardo 213

Saunier Trassy, Julien 3
Serin, Frédéric 98
Shen, Huizhang 87
Siegel, Pierre 33
Sierra, Basilio 118
Stamatatos, Efstathios 77
Su, Sen 108

Tchalakova, Maria 222
Thomas, Kavita E. 150

Wieruchoń, Sławomir T. 245

Yang, Fangchun 108
Yankova, Milena 222
Yao, Ruipu 87
Yao, Yonglei 108
Yu, Yong 138

Zargayouna, Mahdi 3
Zhang, Huajie 138
Zhao, Jidi 87
Zhu, Haiping 138