Lecture Notes in Computer Science 4879

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

Editorial Board

Ernesto Damiani Kokou Yetongnon
Richard Chbeir Albert Dipanda (Eds.)

Advanced Internet Based Systems and Applications

Second International Conference
on Signal-Image Technology
and Internet-Based Systems, SITIS 2006
Hammamet, Tunisia, December 17-21, 2006
Revised Selected Papers

Springer

Volume Editors

Ernesto Damiani
Universitá degli Studi di Milano
Dipartemento Tecnologie dell'Informazione
Via Bramante 65, 26013 Crema, Italy
E-mail: damiani@dti.unimi.it

Kokou Yetongnon
Université de Bourgogne, LE2I-CNRS
Aile de l'Ingénieur, 21078 Dijon Cedex, France
E-mail: kokou.yetongnon@u-bourgogne.fr

Richard Chbeir
Université de Bourgogne, LE2I-CNRS
Aile de l'Ingénieur, 21078 Dijon Cedex, France
E-mail: richard.chbeir@u-bourgogne.fr

Albert Dipanda
Université de Bourgogne, LE2I-CNRS
Aile de l'Ingénieur, 21078 Dijon Cedex, France
E-mail: adipanda@u-bourgogne.fr

Library of Congress Control Number: Applied for

CR Subject Classification (1998): H.3.5, H.3.3, H.3-4, I.2, C.2.4

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

ISSN 0302-9743
ISBN-10 3-642-01349-X Springer Berlin Heidelberg New York
ISBN-13 978-3-642-01349-2 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12652204 06/3180 5 4 3 2 1 0

Preface

In recent years, Internet-based systems and applications have become pervasive and have been the focus of many ongoing research efforts. They range from semi-structured information, to multimedia systems and applications, to P2P and ad hoc information sharing networks and service-centric systems and applications. This book presents a collection of articles from the best papers presented at the SITIS 2006 International Conference, aiming to cover recent advanced research on distributed information systems, including both theoretical and applied solutions.

This volume is designed for a professional audience practitioners and researchers in industry. It is also suitable as a reference or secondary text for advanced-level students in computer science and engineering. The articles in this book are a selection of papers presented at the IMRT and WITDS tracks of the international SITIS 2006 conference.

The authors were asked to revise and extend their contributions to take into account the comments and discussions made at the conference. A large number of high-quality papers were submitted to SITIS 2006, demonstrating the growing interest of the research community for Internet-Based and multimedia information systems.

We would like to acknowledge the hard work and dedication of many people. Our deepest gratitude goes to the authors who contributed their work. We appreciate the diligent work of the SITIS Committee members. We are grateful for the help, support and patience of the LNCS publishing team. Finally, thanks to Iwayan Wikacsana for his invaluable help.

February 2007

Ernesto Damiani
Kokou Yetongnon
Richard Chbeir
Albert Dipanda

Table of Contents

Part I: Query Languages and Information Retrieval

Part II: Multimedia Databases and Systems

Part V: Web Engineering and Services

Part VI: Web Semantics and Semi Structured Data

An Automatic Video Text Detection, Localization and Extraction Approach

Chengjun Zhu, Yuanxin Ouyang, Lei Gao, Zhenyong Chen, and Zhang Xiong

School of Computer Science and Technology, Beihang University
No. 37 Xue Yuan Road, Haidian District, Beijing, P.R.China
{zhucj,gaol}@cse.buaa.edu.cn, {oyyx,chzhyong,xiongz}@buaa.edu.cn

Abstract. Text in video is a very compact and accurate clue for video indexing and summarization. This paper presents an algorithm regarding word group as a special symbol to detect, localize and extract video text using support vector machine (SVM) automatically. First, four sobel operators are applied to get the EM(edge map) of the video frame and the EM is segmented into N×2N size blocks. Then character features and characters group structure features are extracted to construct a 19-dimension feature vector. We use a pre-trained SVM to partition each block into two classes: text and non-text blocks. Secondly a dilatation-shrink process is employed to adjust the text position. Finally text regions are enhanced by multiple frame information. After binarization of enhanced text region, the text region with clean background is recognized by OCR software. Experimental results show that the proposed method can detect, localize, and extract video texts with high accuracy.

Keywords: video text detection, support vector machine(SVM), multilingual texts, video OCR.

1 Introduction

Nowadays, video is the most popular media delivered via TV, internet and wireless network. It leads an increasing demand for systems that can automatically query, search and locate interested content in an enormous quantity of video data. There are two type features[1] that extracted from video data can be used for these goals: low-level and high-level features. Low-level features include object shape, region intensity, color, texture, motion descriptor, audio measurement. High-level include human face detection, speaker identification, character and text recognition. Among all these video features, video text is highly compact and structured, and routinely provides such valuable indexing information such as scene location, speaker names, program introduction, sports scores, special announcements, dates and time. Compared to other video features, information in text is more suitable for efficient video indexing.

In general, video text recognition can be divided into four steps: detection, localization, extraction, OCR recognition. Since the previous three steps generate a binary text image, the recognition can be done by commercial document OCR software. So only the first three steps are discussed in this paper. However, the text detection, localization and extraction in video present many more difficulties than the text

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 1–9, 2009.

segmentation in binary documents due to complex background, degraded text quality and different language characteristics.

Although many methods have been proposed for this task in the last decade[2], [3], [4], [5], [7], [8], few of them address the correlation of video text, especial in the text detection, and many researcher address only contrast, texture and structure feature of single character. But background shape maybe have similar feature, it led to high false detection rate. We observe that the text line layout is different from background objects layout in two-dimension. In our research we regard text group as a special type of symbol that has the union of contrast feature and two-dimension layout feature. Since we use correlation of video text, our method can reduce false detection rate obviously.

Base on related research, our research includes three parts: text detection, localization, and extraction. Because video text stays for a number of consecutive frames, to reduce detection time, As shown in Fig.1, the video shot is sampled every 15 frames. First, edge detector is applied to get the EM(edge map) of the video frame, frequency filter is used to filter low frequency component of EM, then EM character features and characters group structure features are extracted to construct a feature vector. We use a pre-trained SVM to partition each block into two classes: text and non-text blocks. Secondly a dilatation-shrink process is employed to localize the text position. Finally text regions are enhanced by multiple frame information. After binarization of enhanced text region, the text region with clean background was recognized by OCR software.

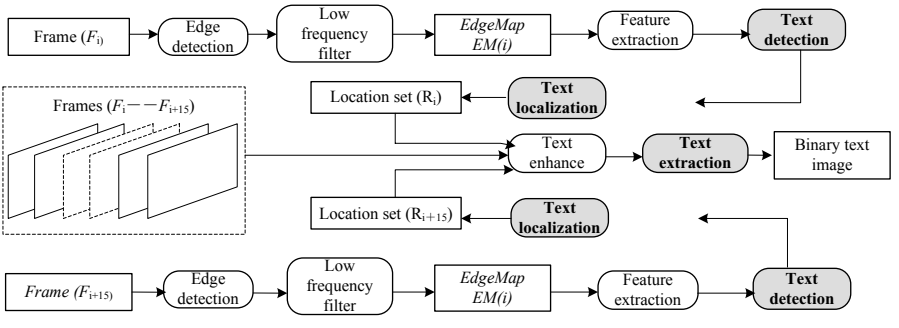


Fig. 1. Algorithm paradigm

2 Text Detection

2.1 SVM Classifier

Among all classifiers, the Support Vector Machine (SVM)[6], [13] has shown promising results owing to its good generalization ability. SVM has good performance for pattern classification problem by minimizing the Vapnik-Chervonenkis dimensions and achieving a minimal structural risk. The idea is finding a hyper-plane that separate the positive example from negative ones in the training set while maximizing distance of them from the hyper-plane. Fig.2 illustrates the maximum margin separating

hyperplane and the support vectors in the two- dimensional space. The norm vector W represents the direction of the separating hyperplane, which is determined through a set of support vector SV. To handle nonlinearly separable classes, a kernel function K is used to map the original data points to a higher dimensional space in which data points are linearly separable. A new sequence d is classified by the sign of the following decision function:

$$f(d) = \sum \lambda_i y_i K(d_i, d) + b, \quad d_i \in SV \ . \tag{1}$$

Where $K(d_i, d)$ is the kernel function, y_i is the class label of the support vector d_i, λ_i is the weight of d_i, and b is the bias. For a SVM with a linear kernel, norm (1) can be simplified as:

$$f(d) = \sum_{i=1}^{n} W_i * X_i + b \ . \tag{2}$$

Where $d=<x_1,x_2, \bullet \bullet \bullet, x_n>$, $w=<w_1,w_2, \bullet \bullet \bullet, w_n>$, and w_i is the weight for ith feature. d is classified as positive if $f(d)>0$ and as negative otherwise. In this case, finding the SVM classifier is determining the w_i and the bias b.

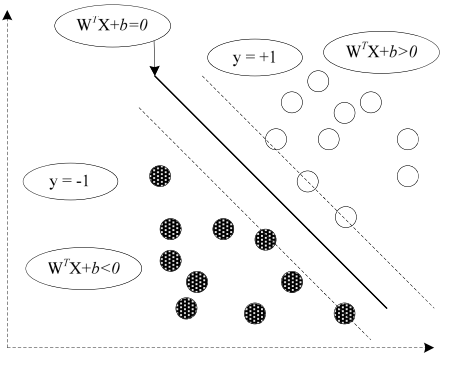


Fig. 2. A linear SVM in two-dimensions space

In fact, video text detection can be regarded as a classification problem and uses SVM to classify video frame region into text region ($f(d)>0$) and non-text region($f(d)<0$).

2.2 EdgeMap

According to stroke statistics, Chinese characters consist of four directional strokes, i.e., horizontal, vertical, up-right-slanting and up-left-slanting, and English characters consist of vertical and horizontal strokes too. So the text region contains rich edge information in the four directions. Correspondingly, we use a set four simple sobel edge filters $S=\{S_h, S_v, S_{rd}, S_{ld}\}$, shown in Fig.3, to filter the video frame data, then get output S_H, S_V, S_{LD} and S_{RD}.

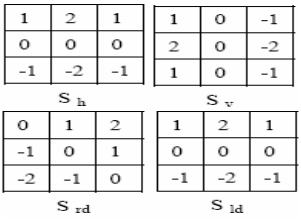


Fig. 3. Four directions Sobel edge filters

Each filtered output is combined to construct EM by the maximum function as follows:

$$EM(i) = MAX(S_H, S_V, S_{RD}, S_{LD}) \ . \tag{3}$$

Because text has high contrast against its background, text edge information is mainly consisted of high frequency components. We apply an adaptive threshold to filter low frequency components. The filter process can be represented by Equation(4):

$$\begin{aligned} &avg(i) = sum(i)/CountNoZer\ o(i) \\ &EM(x, y) == 0 \quad if\ \ EM(x, y) < avg(i) \end{aligned} \ . \tag{4}$$

where $Sum(i)$ represents the sum of gray value of $EM(i)$, $CountNonZero(i)$ represents count of pixels of which the gray value is greater than 0, $avg(i)$ represents the average value. Thus shown Fig. 4, background edge is reduced obviously.

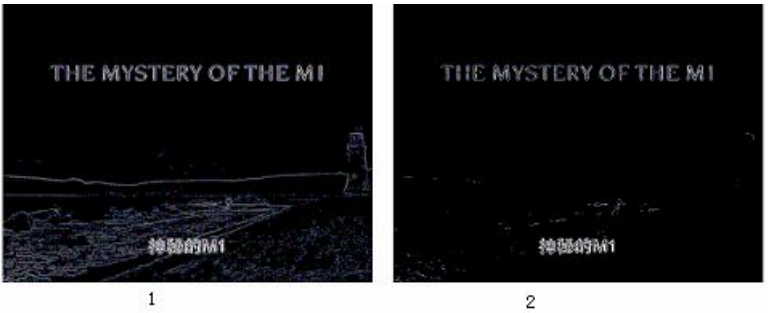


Fig. 4. (1). Video frame'EM. (2).Filtered EM

2.3 Extraction of Feature Vector

If regard a single character as pattern symbol, and ignore the correlation of characters, it will lead to high false detection rate. As shown in Fig.5.(1,2), if not think about context, we cannot know whether it is Chinese character representing "one" or only a horizontal line. At the same time, in the video, single character cannot represent the current content of video, and the union of characters, that is, word can represent the video content well. In Fig.5(3), two Chinese characters represent a role name of a famous Japanese cartoon TV. So when selecting text region feature vector, we choose the comprehensive feature of single character and the feature of context characters.

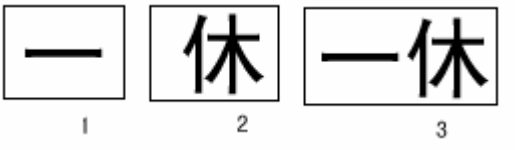


Fig. 5. (1)(2)Single Chinese character, (3)Two Chinese character

To extract feature, $EM(i)$ is divided into blocks B_i and the size of each block is $N\times 2N$. The element of feature vector is calculated by the following two steps.

Step 1: According to symmetry of character structure, block B_i is divided into eight smaller sub-blocks $SubB_{ij}$, $1\le j\le 8$, with same size. Since text region have high contrast against background, then we calculate the mean μ_{ij} and the variance σ_{ij} of $SubB_{ij}$ by

$$\mu_{ij} = \frac{1}{4N^2}\sum_{k=0}^{N/2-1}\sum_{l=0}^{N/2-1} SubB_{ij}(k,l) \ . \tag{5}$$

$$\sigma_{ij} = \frac{1}{2N}\sqrt{\sum_{k=0}^{N/2-1}\sum_{l=0}^{N/2-1}(SubB_{ij}(k,l)-\mu_{ij})^2} \ . \tag{6}$$

Where $subB_{ij}(k,l)$ represents the gray value of $pixel(k,l)$. Then we get first 16($2\times 8=16$) elements of feature vector f_i.

Step 2: According to two-dimension layout feature of characters, we divide block B_i into two sub-block B_{i1}, B_{i2} with size $N\times N$. Within each sub-block, we calculate the centroid coordinate, $G_{i1}(G(x_{i1}), G(y_{i1}))$ and $G_{i2}(G(x_{i2}),G(y_{i2}))$, then we can get the distance from the centroid coordinate to (0, 0) by

$$\begin{aligned}\rho_{i1} &= \sqrt{G^2(x_{i1})+G^2(y_{i1})},\\ \rho_{i2} &= \sqrt{G^2(x_{i2})+G^2(y_{i2})}\end{aligned} \ . \tag{7}$$

The angle θ^i between vector (G_{i1}, G_{i2}) and the horizontal line can be calculated by

$$\theta^i = \arctan\left(\frac{G(y_{i2})-G(y_{i1})}{G(x_{i2})-G(x_{i1})+N}\right) \ . \tag{8}$$

The distance ρ_{i1}, ρ_{i2} and angle θ^i construct last 3($2\times 1+1=3$) elements of the feature vector f_i. Using the SVM classifier, we can separate text region and non-text region easily.

3 Text Region Localization

If video text is related to video content, it must have been presented in a way easy to read with a horizontal orientation. So after separating text region and non-text region, we can combine conjoint text region into a big one. However, the initialing text bounding box may not include all text strokes. As shown in Fig.6.(a), we employ dilatation-shrink process to adjust the position of the text line bounding box.

Step 1: To the bounding box s_i of initialing bounding box set S, if $s_i \cap s_j \neq null$, then set $s_i = s_i \cup s_j$, and delete s_j from bounding box set S. Finally we get a new bounding box set S, and rename it by H. Every element of H is isolated from each other.

Step 2: After step1, we combine conjoint text region, and get the initial position of video text line. Now we employ dilatation-shrink process to adjust bounding box position of text line. To each h_i of H, calculate average value $E(h_i)$of the gray value of all pixels inside bounding box h_i: (a). To every line of h_i, calculating projection number n of pixel whose gray value is greater than $E(h_i)$; (b). If $n>5$, then dilate bounding box and move the line outside by one pixel. Repeat (b), till $n \leq 5$. (c). If $n=0$, then shrink the bounding box and move the line inside by one pixel. Repeat (c), till $n \neq 0$. (d) At last, move every line of bounding box outside by two pixel.

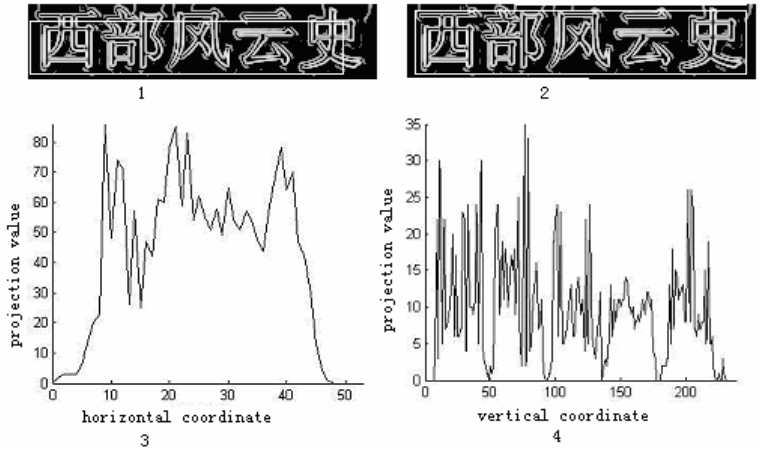


Fig. 6. Projection paradigm (1) Result of detection. (2) Result of localization. (3) Horizontal projection paradigm. (4) Vertical projection paradigm.

In Fig.6(3) and Fig.6 (4), we give horizontal and vertical projection of Fig.6 (1), and the projection value is satisfied with dilatation-shrink process. In Fig.6(2), it can be seen that a good localization result is obtained.

4 Text Enhancement and Extraction

After text detection and localization, we get the text image, but it cannot be put into the OCR software directly, since characters are blended with a complex background.

4.1 Text Enhancement

Video text has the following temporal features: (1). the same text usually appears over several continuous frames. (2). motive characters have a dominant translation direction: linearly horizontal or vertical direction. (3). when text appear or disappear, text region color change significantly in adjacent frames.

According to these temporal features, tang [3] verifies whether the text appearance first time or not. Let C represent video frames set, for a video frame $f_t \in C$ contains text region $r_i(f_t)$. If $r_i(f_t)$ appear first time, calculate $D(r(f_t), r(f_{t+15}))$ by

$$D(r(f_t), r(f_{t+15})) = \iint_{EM_i(f_t)} \left| em_i(l,m) - em_j(l,m) \right| dldm \,. \tag{9}$$

Smaller $D(r(f_t)$ is, $r(f_{t+15}))$, more similar $r_i(f_t)$ is with $r_i(f_{t+15})$. According to the similarity, we can partition the video into frame set $C_i \subset C$. Within C_i, the text stays in the same position. Two type of methods of text enhancement techniques, such as the minimum pixel search [8] and frame averaging method [9], can be employed to remove the complex background. For a frame set C_i, frame $f_t \in C_i$ contains region $r_i(f_t)$. Because the character is stable and the background is changing, the background can be removed by minimum pixel search, which is expressed as

$$\hat{\lambda} = \min_{f_t \in C_i} \gamma_i(f_i) \,. \tag{10}$$

Similarly, the frame average method enhance text by

$$\bar{\gamma} = \frac{1}{|C_i|} \sum_{f_t \in C_i} \gamma_i(f_i) \,. \tag{11}$$

Where $|C_i|$ denotes the number of frame in C_i. Fig.8 (a, b1, b2) shows an example of text enhancement by using these two methods. The complex background is reduced greatly by employing text temporal information. When the background is moving, the minimum pixel method can reduce background well than frame average method.

4.2 Text Binarization

Although we have enhanced text image by integrating multiple frame information, before it is put into OCR software, it must be converted to the binary image which can be accepted by OCR software. It means all pixels of characters are in black and others in white. We uses Otsu[9] threshold method to generate a binary text image. Fig. 8(c1, c2) shows the threshold example of enhanced text image.

5 Experimental Results

No matter whether it is an artificial text or a scene text, if it is related to the video content, its size must be reasonable. In other words, the size of character cannot be too small or too big. So in our experiment we consider characters with size 15×15 pixels at least. The moving window N×2N must contain main features of characters. So window N×2N must contain two characters, and we set N=20 pixels length. Because there are not standard test video sequence about video text recognition, we select training and test material from news and movies video shots which contain text in simple or complex background, moving linearly or staying stable. We design a tool named SuperEye-Trainer, which can divide video frame into N×2N blocks and label the block, then we get a series of training data <f, y>, where f represents the feature vector and $y \in \{-1, +1\}$. In the experiment we must assign parameter λ of kernel

function of SVM, usually $\lambda=1/k$, where k represents the dimension of feature vector. Here we set $k=19$.

Hua[12] proposed an objective and comprehensive performance evaluation protocol for video text detection algorithms. Among his criterion, we choose detection number, false detection number, detection rate, and false detection rate to evaluate our text detection algorithm. Table 1 and Fig.7 show the detection results. It can be seen that the detection results are rather good.

Table 1. Text detection result

Train data (frame)		400
Test data number(block)	Text region	200
	Background region	200
Detection number		184
False detection		8
Detection rate		92%
False detection rate		4%

Fig. 7. Text region detection result

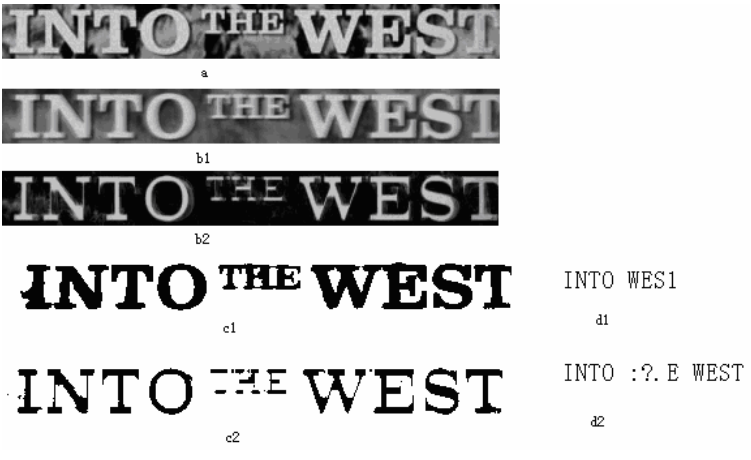


Fig. 8. (a). Original video frame. (b1). Frame average result. (b2). Minimum pixel search result,. (c1). Binary image of b1. (c2). Binary image of 2. (d1). Recognition result of c1, (d2). Recognition result of c2.

For the extracted characters, we put them into Microsoft office document imaging OCR module. The recognition rate achieves to 82%. Fig. 8(d1, d2) shows the recognition result, and it represents the current video content obviously.

6 Conclusion

Only the union of characters can represent current video content well. Based on this point, we develop an automatic video-text detection and recognition system. Using pre-training SVM classifier to detect video text region, the system detect video text with high precision and efficiency. Experimental result shows that our method can detect the text region by a high accuracy rate. Furthermore, aided by a text image enhance and binarization methods, we improve video OCR accuracy to 82%.

References

1. Aslandogan, Y.A., Yu, C.T.: Techniques and systems for image and video retrieval. IEEE Trans. Knowledge Data Eng. 11, 56–63 (1999)
2. Lyu, M.R.: Jiqiang Song; Min Cai: A comprehensive method for multilingual video text detection, localization, and extraction. IEEE Transactions on Circuits and Systems for Video Technology 15(2), 243–255 (2005)
3. Tang, X., Gao, X., Liu, J., et al.: A Spatial-Temporal Approach for Video Caption Detection and Recognition. IEEE Trans On Neural Networks, 961–971 (2002); special issue on Intelligent Multimedia Processing
4. Zhang, H.J.: Content-based video analysis, retrieval and browsing. Microsoft Research Asia, Beijing (2001)
5. Chen, D., Bourlard, H., Thiran, J.-P.: Text Identification in Complex Back-ground Using SVM. In: CVPR 2001, vol. II, pp. 621–626 (2001)
6. Vapnik, V.: The Nature of Statistical Learning Theory 361, 581–585 (1996)
7. Sato, T., Kanade, T., Kughes, E.K., Smith, M.A., Satoh, S.: Video OCR: Indexing digital news libraries by recognition of superimposed captions. ACM Multimedia Syst (Special Is-sue on Video Libraries) 7(5), 385–395 (1999)
8. Li, H.P., Doemann, D., Kia, O.: Text extraction, enhancement and OCR in digital video. In: Proc. 3rd IAPR Workshop, Nagoya, Japan, pp. 363–377 (1998)
9. Otsu, N.: A Threshold Selection Method from Grey-Level Histograms. IEEE Trans. Systems, Man, and Cybernetics 9(1), 377–393 (1979)
10. Song, J., Cai, M., Lyu, M.R.: A robust statistic method for classifying color polar-ity of video text. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003), April 2003, vol. 3, pp. 581–584 (2003)
11. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
12. Hua, X.-S., Wenyin, L., Zhang, H.-J.: Automatic Performance Evaluation for Video Text Detection, icdar. In: Sixth International Conference on Document Analysis and Recognition (ICDAR 2001), p. 0545 (2001)
13. Zhou, S., Wang, K.: Localization site prediction for membrane proteins by integrating rule and SVM classification. IEEE Transactions on Knowledge and Data Engineering 17(12), 1694–1705 (2005)

Towards an MPEG-7 Query Language

Mario Döller[1], Harald Kosch[1], Ingo Wolf[2], and Matthias Gruhne[3]

[1] Department of Distributed Information Technology, University Passau
{Mario.Doeller,Harald.Kosch}@uni-passau.de
[2] Department Platforms for Media Broadband, T-Systems Enterprise Services GmbH
WolfI@t-systems.com
[3] Institut Digitale Medientechnologie (IDMT), Fraunhofer Illmenau
ghe@idmt.fraunhofer.de

Abstract. Due to the growing amount of digital media an increasing need to automatically categorize media such as music or pictures has been emerged. One of the metadata standards that has been established to search and retrieve media is MPEG-7. But it does not yet exist a query format that enables the user to query multimedia metadata databases. Therefore the MPEG committee decided to instantiate a call for proposal (N8220) for an MPEG-7 query format (MP7QF) framework and specified a set of requirements (N8219). This paper introduces a MP7QF framework and describes its main components and associated MP7QF XML schema types. The framework makes use of the MPEG-21 digital item declaration language (DIDL) for exchanging MP7QF Items along various MP7QF frameworks and client applications. An MP7QF Item acts as container for the input query format and output query format of a user query request.

This paper concentrates on components of the framework such as session management, service retrieval and its usability and excludes consciously definitions and explanations of the input and output query format.

1 Introduction

Stimulated by the ever-growing availability of digital audiovisual material to the user via new media and content distribution methods an increasing need to automatically categorize digital media has emerged. Descriptive information about digital media which is delivered together with the actual content represents one way to facilitate this search immensely. The aims of so-called metadata ("data about data") are to e.g. detect the genre of a video, specify photo similarity or perform a segmentation on a song, or simply recognize a song by scanning a database for similar metadata. A standard that has been established to specify metadata on audio-visual data is MPEG-7 and has been developed by MPEG. This organization committee also developed the successful standards known as MPEG-1 (1992), MPEG-2 (1994) and MPEG-4 (version 2 in 1999). MPEG-7 had a broad impact to experts of various domains of multimedia research. For instance, there are several proposals for storing MPEG-7 descriptions in multimedia databases (e.g., MPEG-7 MMDB [6]). As MPEG-7 bases on XML Schema, one distinguishes research in native XML databases (e.g., Tamino [15]) and on mapping strategies for (object)-relational databases (e.g., Oracle [12]). Then, there are several multimedia applications supporting and using MPEG-7. To mention only a few: VideoAnn [18], Caliph and

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 10–21, 2009.

Emir [10] etc. Finally, many document query languages such as [SDQL 96], [XML-QL 99], [Lorel 00], [YATL 98], [XQL 98], recent W3C [XQuery], etc., have been proposed for XML or MPEG-7 document retrieval. However, these languages cannot adequately support MPEG-7 description queries mainly due to two reasons. Either, they do not support query types which are specific for retrieving multimedia content such as query by example, query based on spatial-temporal relationships, etc. Or, there is no standardized interface defined and each query language or MPEG-7 database offers its own query interface, which prevents clients experiencing aggregated services from various MPEG-7 databases.

At the 77^{th} MPEG meeting in July 2006, the MPEG committee decided to instantiate a call for proposal (N8220 [1]) for a MPEG-7 query format (MP7QF) framework and specified a set of requirements (N8219). The objective of the MP7QF framework is to provide a standardized interface to MPEG-7 databases allowing the multimedia retrieval by users and client applications, based on a set of precise input parameters for describing the search criteria and a set of output parameters for describing the result sets.

In this context, we briefly analyze currently available query languages and approaches for multimedia and XML data in consideration of the introduced MP7QF requirements and propose a new architecture for a MP7QF framework. This paper concentrates on components of the framework and its usability and excludes consciously definitions and explanations of the input and output query format. The query format will be introduced in a separate paper.

The remainder of this paper is organized as follows: Section 2 specifies requirements for an MP7QF framework. Then, Section 3 deals with related work to MPEG-7 query languages and databases. The MP7QF framework is described in Section 4 which is divided in 4 subsections namely, subsection 4.1 describing a multimedia retrieval scenario using the proposed framework, subsection 4.2 presenting the architecture and its components, subsection 4.3 specifying the MP7QF Item and subsection 4.4 dealing with the retrieval of MP7QF-Interpreter. The MP7QF-Interpreter deals as adaptor for MPEG-7 databases and is responsible for transforming MP7QF queries into queries of the query language of the respective database. Finally, this paper is summarized in Section 5.

2 Requirements for an MP7QF Framework

In general, one can distinguish between two main types of requirements: requirements for the framework and requirements for the query language. In the following, selected requirements are explained in more detail. A complete list is given in MPEGs requirement paper (N8219).

2.1 Requirements for the Framework

1.) *Allowing simultaneous search in multiple databases:* The framework should support the distribution of a single query to multiple search engines.

[1] see `http://www.chiariglione.org/mpeg/working_documents/explorations/mp7_qf/mp7_qf_reqs.zip`

2.) *Querying database capabilities:* The framework should allow the retrieval of database capabilities such as supported media formats, supported MPEG-7 descriptors/ descriptor schemes, supported search criteria, etc.

2.2 Requirements for the Query Language

1.) *Query Types*
   a.) *Query by description:* Here, the query language should provide means for the retrieval based on textual descriptions (e.g., names) as well as the retrieval based on desired MPEG-7 descriptions and/or description schemes.
   b.) *Query by example:* The query language should support the retrieval based on representative examples of the desired content.
   c.) *Multimodal query:* The query language should provied means for combining the retrieval based on different media types.
   d.) *Spatial-temporal query:* The query language should support retrieval based on spatial and/or temporal relationships (e.g., Search for images where a red ferrari is in front of a white house.)
2.) *Specifying the result set:* The query language should provide means for specifying the structure as well as the content in the sense of desired data types.
3.) *Querying based on user preferences and usage history:* The query language should consider user preferences and usage history for retrieval.
4.) *Specifying the amount and representation of the result set:* The query language should provide means for limiting the size as well as supporting paging of the result set.

3 Related Work to MPEG-7 Query Languages and Databases

A very good overview of the usability of XML databases for MPEG-7 is provided by Westermann and Klas in [20]. The authors investigated among others the following main criteria: *representation of media descriptions* and *access to media descriptions*. To summarize their findings, neither native XML databases (e.g., Tamino [15], Xindice [16]) nor XML database extensions (e.g., Oracle XML DB [12], Monet XML [14], etc.) provide full support for managing MPEG-7 descriptions with respect to their given requirements. Based on their various limitations (e.g., supported indexing facilities for high-dimensional data), the retrieval capabilities for multimedia data are restrictive.L

In the following, we will have a closer look to the used query languages in various approaches and evaluate them on the requirements presented in Section 2.

XPath [2] (XML Path Language) is a recommendation of the W3C consortium that enables the access to individual parts of data elements in the XML document. In general, an XPATH expression consists of a path (where the main difference to filenames or URIs is that each step selects a set of nodes and not a single node) and a possible condition that restricts the solution set. The main disadvantage of XPATH expression is their limited usability in querying XML documents. For instance, it does not provide means for grouping or joins. In addition, XPath on its own provides no means for querying multimedia data in MPEG-7 descriptions based on the presented criteria.

XQuery [19] is a declarative query language and consists of the following primary areas: The main areas find their counterparts in SQL. For instance, the *for/let* clauses represent the SQL SELECT and SET statements and are used for defining variables respectively iterating over a sequence of values. The *where* clause complies to the SQL WHERE statement by filtering the selection of the *for* clause. The *order-by* finds their analogous in the SQL SORT BY statement and provides an ordering over a sequence of values. The *return* clause respectively SQL RETURN uses a custom formatting language for creating output. A main part of XQuery is the integration of XPath 2.0 and their functions and axis model which enables the navigation over XML structures. Additional parts provide the ability to define own functions analogous to SQL stored procedures and the handling of namespaces. With regard to our requirements, XQuery does not provide means for querying multiple databases in one request and does not support multimodal or spatial/temporal queries.

SQL/XML [3] is an extension of SQL and was developed by an informal group of companies, called SQLX[2], including among others IBM, Oracle, Sybase and Microsoft. A final draft has been standardized as SQL part 14 (SQL/XML) by ANSI/ISO in 2003. Its main functionality is the creation of XML by querying relational data. For this purpose, SQL/XML proposes three different parts. The first part provides a set of functions for mapping the data of (object-) relational tables to an XML document. The second part specifies an XML data type and appropriate functions in SQL for storing XML documents or fragments of them within a (object-) relational model. The third part describes mapping strategies of SQL data types to XML Schema data types. SQL/XML supports the requirement for specifying the representation and content of the result set but on its own (based on its alliance to SQL) there are no means for supporting multimedia retrieval in combination with MPEG-7 descriptions.

The authors in [9] propose an XML query language with multimedia query constructs called MMDOC-QL. MMDOC-QL bases on a logical formalism path predicate calculus [8] which supports multimedia content retrieval based on described spatial, temporal and visual data types and relationships. The query language defines four main clauses: OPERATION (e.g.: generate, insert, etc.) which is used to describe logic conclusions. The PATTERN clause describes domain constraints (e.g., address, etc.). Finally, there exist a FROM and CONTEXT clause which are paired together and can occur multiple times. The FROM clause specifies the MPEG-7 document and the CONTEXT is used to describe logic assertions about MPEG-7 descriptions in path predicate calculus. Of all presented query languages, MMDOC-QL fulfills best the presented requirements. Nevertheless, there are several drawbacks such as simultaneous searches in multiple databases or the integration of user preferences and usage history which are not considered in MMDOC-QL.

XIRQL [4] is a query language for information retrieval in XML documents and bases on XQL [13]. The query language integrates new features that are missing in XQL such as weighting and ranking, relevance-oriented search, data types and vague predicates and semantic relativism. A similar weighting and relevance approach has been introduced in [17].

[2] `http://www.sqlx.org`

Besides, there exist several query languages explicitly for multimedia data such as SQL/MM [11], MOQL [7], $POQL^{MM}$ [5] etc. which are out of scope of this paper based on its limitation in handling XML data.

4 Approach

4.1 Scenario

The following scenario describes a simple interaction process of a user (or client application) with our MP7QF framework during a content-based multimedia retrieval.

Scenario: Query by example with one database. This scenario deals with a query by example request where one image database is included. The query considers the following two low level feature descriptors ScalableColorDescriptor and DominantColorDescritpor for retrieval. In the following, the separate steps are explained in detail:

1. *Connection to the MP7QF Framework:* During the connection process of a user to the MP7QF framework, a new session is established. A session administrates a session id, the connected MP7QF-Interpreter, user preferences and user history. In this context, the user can set its user preferences (as MPEG-7 description) for the established session.
2. *Database selection:* The next step is the selection of desired MP7QF-Interpreter which the framework should consider for retrieval. In this scenario, the user selects the database by its own (see service retrieval scenario presented in Figure 3(a)). Several different connection approaches are described in Section 4.4.
3. *Formulate the query:* This can be realized in combination with a graphical editor (maybe selecting desired descriptors from a tree) or by a command shell. The outcome of this step is an XML instance document based on the input query format (IQF) XML Schema.
4. *Formulate Result Set:* Formulate how the result set should look like (structure and content). The basis for this process is the output query format (OQF) XML Schema.
5. *Transmit Query:* Transmit the query (IQF and OQF instance document) to the MP7QF-Interpreter which is responsible for the transformation of the input query format to the respective target database query language.
6. *Receive Result:* The MP7QF-Interpreter returns the result based on the OQF to the MP7QF framework where its validity is checked. Then the result is forwarded to the user. In this stage, the MP7QF framework extracts user history information and stores the result in the users session information.

4.2 Proposed Architecture

The proposed MP7QF Architecture presented in Figure 1 comprises the main components for fulfilling most of the requirements described in MPEGs requirement paper (N8219) and enables the demonstrated scenario in a standardized way. In the following, the components are described in detail whereas the authors concentrate on components of the framework and its usability and excludes consciously definitions and explanations of the input and output query format. The query format will be introduced in a separate paper.

– **Session Management:** The session management provides means for the storage of user information such as user preferences and user history. In addition, the session management stores the query result for allowing relevance feedback (requirement 4.4.3 of N8219) and the search within the result set of previous searches (requirement 4.4.4 of N8219). For this purpose, our MP7QF proposal introduces a SessionType (see Appendix A) which contains the following elements:

  1.) *ActiveConnection:* The active connection element stores all currently active connections of type ConnectionType. The connections are manipulated (added, deleted) by the use of the Service Management tool. Whenever a retrieval operation is initiated (by calling the Search method), the distributor tool forwards the query to all active connections.
  2.) *QueryResultCache:* The query result cache element stores the result (in form of an OQF type) of the last successful executed retrieval operation. This cache will be used to allow relevance feedback and the search within the result set of previous searches.
  3.) *UserDescription:* The user description complies with the MPEG-7 UserDescription type and manages user data, their preferences and their history. The user preferences and history can be used during the retrieval process to personalize the query (e.g., adding some specific filter operations, etc.).

  An instance of the SessionType is established and maintained for every active user by the session management tool. For this purpose, the session management tool provides the following methods:

  - *SessionID createSession (UserDescription):* This method allows the establishment of a new session for a user. The input parameter contains available user description and corresponds to the MPEG-7 UserDescription descriptor. We suggest that at least the user element is filled. After a successful execution of the operation the created session ID is responded. This session ID applies during the retrieval process for identifying the participating session.
  - *UserDescription closeSession (SessionID):* This method closes an active session which is identified through the given session ID. In addition, the user description is returned which may contain updated history data.
  - *UserDescription closeSession (UserDescription):* Same as before, this method closes an active session. The session is identified through the user description type and here especially by the user element. The main difference to the previous method is, that in this case all open sessions of the respective user are closed.
  - *OQF mp7Search (SessionID, MP7QF Item):* The search method initiates the retrieval process and takes as input parameter the responsible session and the query formulated by IQF and OQF stored in a MP7QF item (see Section 4.3). After completion, the method returns the result as XML document based on the given OQF.

– **Service Management:** This component manages two parts. First, it provides methods for connecting MP7QF-Interpreter to active user sessions. Second, the service management gathers all active MP7QF-Interpreter their ServiceCapabilityDescriptors. For this purpose, every MP7QF-Interpreter must provide such a possibility,

e.g., *ServiceDescriptor getServiceDescription ()*. The management of connections can be realized by using one of the following methods:

- *connectService (SessionID, <vector> ServiceDescriptor):* This method corresponds to the described service retrieval scenario in Figure 3(b) and adds all given MP7QF-Interpreter to an internal list hold at the session management. The ServiceDescriptor contains basic service information such as Host ID, connection information, etc. (see Section 4.4 and Appendix A for a detailed description).
- *<vector> ServiceDescriptor searchService (ServiceCapabilityDescriptor, Boolean):* This method filters the list of registered MP7QF-Interpreter based on a given ServiceCapabilityDescriptor. The ServiceCapabilityDescriptor is described in more detail in Section 4.4. The second parameter specifies a Boolean value which determines whether the filter process should be restrictive meaning that the result set only contains MP7QF-Interpreters which match the retrieval criteria to 100 percent. Otherwise the result set would also contain MP7QF-Interpreters which do not match the criteria for 100 percent (e.g., a specific low level descriptor is missing, a specific IQF operation is not supported, etc.).
- *releaseService (SessionID, <vector> ServiceDescriptor):* This method releases the connected MP7QF-Interpreter from the defined session.

– **Aggregator:** The aggregator is used for combining the result sets of different databases.
– **Distributor:** The distributor splits the user request into calls specific to a certain database. Note: The aggregator and the distributor are only necessary when more than one database is involved in a query.
– **Input Query Format:** The Input Query Format (IQF) specifies the syntax and structure of an MP7 query.
– **Output Query Format:** The Output Query Format (OQF) specifies the syntax and structure of the MP7 query result set.
– **Query Composer:** The Query Composer defines an overall syntax and structure which combines IQF and OQF elements to one request. This request is described and transmitted with the MP7QF Item to respective MP7QF-Interpreter. The Query Composer is also used to assemble the OQF response .
– **Query Interpreter:** The MP7QF Items are transformed into specific bindings (e.g., XQuery, SQL, etc.) of the target databases. The result set is handled by the Query Composer to produce the OQF response.

4.3 MP7QF Item

The MPEG-7 Query Format Item (MP7QF Item) is used for the exchange of MP7 query information between MP7 management tools on different machines and can also be used by the client for interacting with the MP7QF framework. The description of the MP7QF Item bases on the MPEG-21 digital item declaration language (DIDL) standard [1]. The architecture of the MP7QF Item (see Figure 2) consists of the following elements:

– *MP7QF ConnDescriptor:* This descriptor represents the connection information/ session object information.

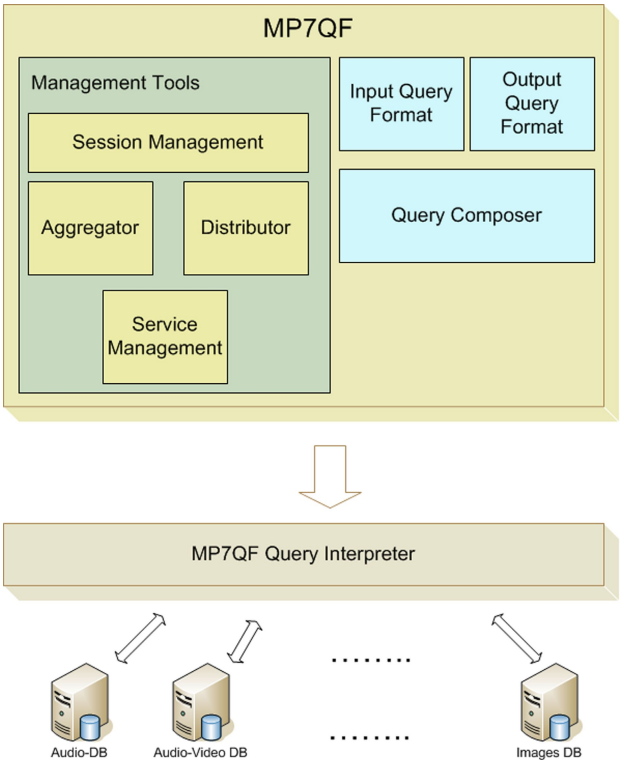


Fig. 1. MPEG-7 Query Format Framework Architecture

- *Query Input Item:* The Item contains a set of Descriptors, including MPEG-7 DescriptionUnits, the actual query and optionally on or more components (in case of query by example).
- *Query Output Item:* The Item contains a Result set Item and optionally an Item carrying presentation information, e.g. an XSLT

An MP7QF Input Item and corresponding output item is stored in a container.

4.4 MP7QF-Interpreter Retrieval

In order to query MPEG-7 databases, we have to introduce an MP7QF-Interpreter which serves as adaptor of the MP7QF framework to available MPEG-7 databases. Here we have to note, that every MPEG-7 database type (e.g., native XML-database, or OR SQL-database, etc.) must provide such an interpreter. During query execution, the MP7QF-Interpreter is responsible for transforming the MP7QF Item into database specific calls. One possible binding of the MP7QF-Interpreter would be the use of Web-Service technology which is explicitly mentioned by the MPEG requirements.

The MP7QF framework supports two different MP7QF-Interpreter retrieval scenarios. First, as displayed in Figure 3(a), the user has the possibility to connect to any

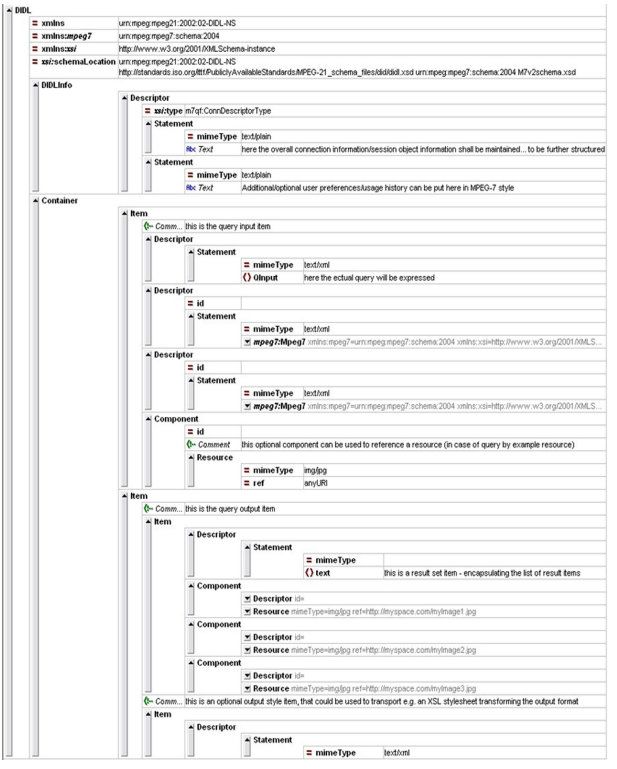


Fig. 2. MPEG-7 Query Format Item

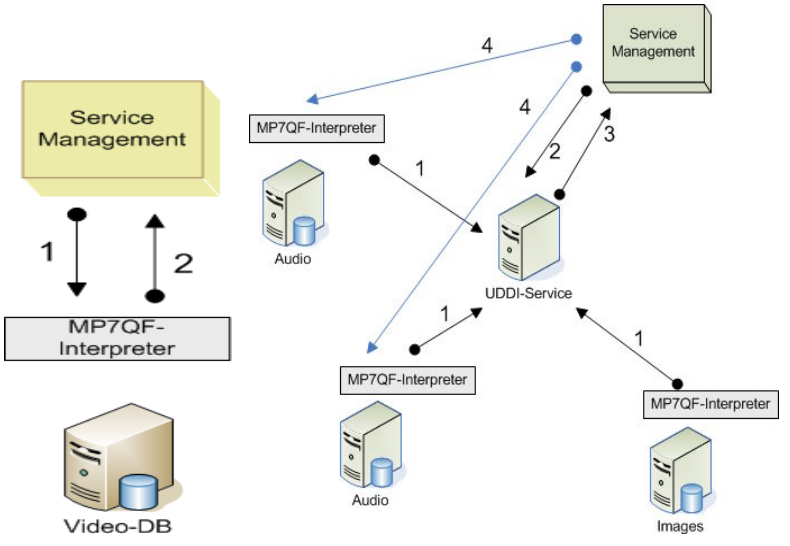
desired database. Second, as shown in Figure 3(b), one can select several databases based on its stored media type (e.g., audio) and by their provided MPEG-7 Profile, IQF query types and operations.

In the following, the different steps of both scenarios are explained in detail:

– **Manual Service Retrieval**
  1. In a first step, the service management component connects to the MP7QF-Interpreter based on a fixed URL given by the user. This is realized by calling the *connectService* method of the service management component.
  2. During this step, the MP7QF-Interpreter transmits its supported service capabilities and connection information in form of the ServiceDescription type to the service management tool. This information is stored for the currently active session (SessionType) in the session management.

– **Automatic Service Retrieval**
  1. First, every participating media database has to register at an exclusive server (UDDI Service [3] in case of Web-Service binding). During this registration all MP7QF-Interpreters send their ServiceDescriptions.

[3] Universal Description, Discovery and Integration (UDDI) see `http://www.uddi.org/`



(a) Manual Service Retrieval **(b)** Automatic Service Retrieval

Fig. 3. Service Retrieval Approaches

2. Then, the service management component of the MP7QF framework queries the UDDI Service based on the user needs. This query can contain requests based on supported MPEG-7 Profiles or IQF query types and operations and is represented by an instance of service capability type.
3. The UDDI Service responds a list of databases fitting the required constraints.
4. The MP7QF framework connects to the proposed databases and receives their capabilities

ServiceDescriptorType. The ServiceDescriptor type combines all information about one MP7-Interpreter and its database. In detail, it includes the connection interface described by the ConnectionType, and the supported capability of the MP7QF-Interpreter and its database in form of the ServiceCapability type. The type is used as either input or output parameter of the following methods: *connectService, searchService* and *releaseService*.

ServiceCapabilityType. The ServiceCapability Type describes information of a MP7QF-Interpreter concerning its supported query types and the underlying MPEG-7 profile. A MP7QF-Interpreter represents the interface to an MPEG-7 database and is responsible for transforming incoming MP7QF Items containing IQF and OQF requests to the respective query language of the target database (e.g., XQuery). Due to the diversity of the MPEG-7 standard it is very likely that an MPEG-7 database only supports a subset of available MPEG-7 descriptors. In series, this is true for the IQF data types and operators. For this purpose, the ServiceCapability Descriptor defines the following elements:

– *SupportedProfile:* This element specifies the MPEG-7 profile the target database provides.

- *SupportedIQFTypes:* This element contains a list of supported input query format types the target database is able to process.
- *SupportedIQFOperations:* This element provides a list of supported input query format operations the target database is able to evaluate.
- *UsageCondition:* This element contains a set of predefined usage conditions such as free of charge, authentication required, payed service etc.

The ServiceCapability Type is used during a service retrieval process in two different cases: First, it describes the capabilities of an MP7QF-Interpreter and the database. Second, a user can formulate its desired requirements an MP7QF-Interpreter must support.

5 Summarization

This paper introduced an MPEG-7 query format (MP7QF) framework based on the defined requirements in the 77^{th} MPEG meeting in July 2006. The framework provides means for session and service management which have been described in detail. In addition, parts of our MP7QF XML Schema have been introduced such as ServiceDescription, ServiceCapabilityDescriptor, SessionType, etc. Nevertheless, it has to be noted that this paper concentrated on components of the framework such as session management and service retrieval and its usability and excludes consciously definitions and explanations of the input and output query format.

References

1. Bormans, J., Hill, K.: Overview of the MPEG-21 standard. ISO/IEC JTC1/SC29/WG11/N5231 (October 2002)
2. Clark, J., DeRose, S.: XML Path Language (XPath). W3C Recommendation (1999), `http://www.w3.org/TR/xpath`
3. Eisenberg, A., Melton, J.: SQL/XML is Making Good Progress. ACM SIGMOD Record 31(2), 101–108 (2002)
4. Furh, N., Grossjohann, K.: XIRQL: A Query Language for Information Retrieval in XML Documents. In: Proceedings of the 24th ACM-SIGIR Conference on Research and Development in Information Retrieval, New Orleans, Louisiana, USA, pp. 172–180 (2001)
5. Henrich, A., Robbert, G.: POQL^{MM}: A Query Language for Structured Multimedia Documents. In: Proceedings 1st International Workshop on Multimedia Data and Document Engineering (MDDE 2001), pp. 17–26 (July 2001)
6. Kosch, H., Döller, M.: The MPEG-7 Multimedia Database System (MPEG-7 MMDB). Journal of Systems and Software (accepted for publication) (in Press by Elsevier) (to appear in spring 2007)
7. Li, J.Z., Özsu, M.T., Szafron, D., Oria, V.: MOQL: A Multimedia Object Query Language. In: Proceedings of the third International Workshop on Multimedia Information Systems, Como Italy, pp. 19–28 (1997)
8. Lui, P., Charkraborty, A., Hsu, L.H.: Path Predicate Calculus: Towards a Logic Formalism for Multimedia XML Query Language. In: Proceedings of the Extreme Markup Languages, Montreal, Canada (2000)

9. Lui, P., Charkraborty, A., Hsu, L.H.: A Logic Approach for MPEG-7 XML Document Queries. In: Proceedings of the Extreme Markup Languages, Montreal, Canada (2001)
10. Lux, M., Klieber, W., Granitzer, M.: Caliph & Emir: Semantics in Multimedia Retrieval and Annotation. In: Proceedings of the 19th International CODATA Conference 2004: The Information Society: New Horizons for Science, Berlin, Germany, pp. 64–75 (2004)
11. Melton, J., Eisenberg, A.: SQL Multimedia Application packages (SQL/MM). ACM SIGMOD Record 30(4), 97–102 (2001)
12. Murthy, R., Banerjee, S.: XML Schemas in Oracle XML DB. In: Proceedings of the 29th VLDB Conference, Berlin, Germany, pp. 1009–1018. Morgan Kaufmann, San Francisco (2003)
13. Robie, J.: XQL (XML Query Language) (1999), `http://www.ibiblio.org/xql/xql-proposal.html`
14. Schmidt, A., Kersten, M.L., Windhouwer, M., Waas, F.: Efficient relational storage and retrieval of XML documents. In: Suciu, D., Vossen, G. (eds.) WebDB 2000. LNCS, vol. 1997, p. 137. Springer, Heidelberg (2001)
15. Schning, H.: Tamino - a DBMS designed for XML. In: Proceedings of the 17th International Conference on Data Engineering (ICDE), pp. 149–154 (April 2001)
16. Staken, K.: Xindice Developers Guide 0.7. The Apache Foundation (December 2002), `http://www.apache.org`
17. Theobald, A., Weikum, G.: Adding Relevance to XML. In: Suciu, D., Vossen, G. (eds.) WebDB 2000. LNCS, vol. 1997, pp. 35–40. Springer, Heidelberg (2001)
18. Tseng, B.L., Lin, C.-Y., Smith, J.R.: Video Personalization and Summarization System. In: Proceedings of the SPIE Photonics East 2002 - Internet Multimedia Management Systems, Boston, USA (2002)
19. W3C. XML Query (XQuery). W3C (2006), `http://www.w3.org/TR/xquery/`
20. Westermann, U., Klas, W.: An Analysis of XML Database Solutions for the Management of MPEG-7 Media Descriptions. ACM Computing Surveys 35(4), 331–373 (2003)

Characterizing Multimedia Objects through Multimodal Content Analysis and Fuzzy Fingerprints

Alberto Messina[1], Maurizio Montagnuolo[2,⋆], and Maria Luisa Sapino[2]

[1] RAI Centro Ricerche e Innovazione Tecnologica, Torino, Italy
a.messina@rai.it
[2] Università degli Studi di Torino, Dip. di Informatica, Torino, Italy
{montagnuolo,mlsapino}@di.unito.it

Abstract. In this paper we introduce a new approach to multimedia data semantic characterisation and in particular television programmes fingerprinting, based on multimodal content analysis and fuzzy clustering. The definition of the fingerprints can be seen as a *space transformation process*, which maps each programme description from the surrogate vector space to a new vector space, defined through a fuzzy clustering method. The fuzzy fingerprint model is well suited for similarity based information retrieval, and it captures "semantic" similarities coming from common pattern in the programme data, at different semantic levels.

Keywords: fuzzy clustering, semantic characterisation, multimodal content analysis, broadcast, multimedia fingerprint.

1 Introduction

A big amount of multimedia content, including images, audio clips and videos, is now available in digital libraries. Efficient and low-cost solutions for semantic characterisation of multimedia content are needed, to enable effective access to these data. In this paper we propose a fingerprinting method that dynamically and automatically derives the *genre* of a television programme from the available data. In particular, we focus on the *television broadcasting domain*, where (i) *users retrieving multimedia data are almost never the ones who took part in the multimedia data production.* Thus, the information about the programme needs either to be directly stored as metadata or automatically inferrable by the system; (ii) *users are often interested in partial retrieval of content.* Therefore, the retrieval architecture should allow *semantic segmentation* of the data, and production of programmes in which parts of the original contents are reused.

The fundamental assumption in this domain is that *multimedia producers very often work according to some common stylistic patterns.* They use specific and recurring aggregations of low-level perceivable features (e.g. colours, editing effects), of structural properties (e.g. rhythm in the sequence of shots) and

⋆ PhD student, supported by EuriX S.r.l., Torino, Italy. — www.eurixgroup.com

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 22–33, 2009.

of cognitive properties (e.g. shots representing softly smiling people, or natural landscapes) to communicate some high-level concepts (e.g. peace and relaxation). Concepts are further combined, at a higher level, to produce more general concepts, which can result in the genre of a video. Under this assumption, our *goal* is to *learn* structural characterisations of multimedia objects from representative sets of programmes and to rely on this characterisation for a similarity based retrieval process. The challenges we have to deal with in our specific application domain generalise to many other domains, and we can comfortably state that our architecture properly fits with other multimedia contexts as well.

As opposed to more traditional multimedia classification approaches, we implement *semantic characterisation* through a process of *aggregation of multimedia objects sharing similar perceptual and structural properties*, rather than directly associating such properties to concepts and classes on the basis of subjective and predefined taxonomies. For this purpose, we collected more than 100 hours of television programmes. In order to preserve compatibility with established television genres, each programme is annotated according to *seven semantic categories*, each of them corresponding to a traditional video genre (Newscasts, Commercials, Cartoons, Football, Music, Weather Forecasts, and Talk Shows). Note that, *this classification is only used as an initial reference*, to help organise the database (and eventually validate our theory against a plain case). We believe that this approach better simulates the human understanding of the semantics that are implied in multimedia production.

1.1 Related Work

Fischer et al. [8] analysed the well defined rules used by video editors (e.g. shot lengths, or volume of audio tracks) to derive stylistic patterns that reflect the producer's intentions. In [24], the genre classifier is based on a C4.5 Decision Tree [15] and a 10-dimensional feature vector representing the visual content of the video. [4] uses Hidden Markov Models (HMMs) [16] to categorise videos. [26] investigates automated video genre classification, using compact representations of MPEG-7 audiovisual descriptors. In [21], camera motion parameters are employed to discern between different kinds of sport. More recent approaches use classification rules from low/mid level features, according to a predefined ontology of the content domain. In [3], an ontology infrastructure for semantic annotation of objects in car racing and football sports video is presented. [2] presents an annotation engine that uses reasoning algorithms to automatically annotate and retrieve events in football video. A rule-based system for automatic annotation of videos is presented in [5].

Despite the number of existing efforts, the problem of comprehensively discerning multimedia genres has not been completely solved yet. In fact, it is difficult to analytically define the *concept of genre*, and any attempt at explicitly defining it ends up being *subjective and domain dependent*. Thus, most of the existing approaches either attempted to discern only few weakly defined genres based on a simple taxonomy [4,5,8,24,26], or only focussed on one specific domain, such as sport videos [2,3,21]. In addition, they use crisp classifiers, assigning an absolute

class label to each multimedia object, although, in the real world scenario, a single object can belong to none, one or more classes at the same time, and although the distinction between different classes is not necessarily sharp. To overcome these limitations we use fuzzy classifiers, which capture additional information about the certainty degree of the classifier decision, and introduce a more natural description of the semantic classification problem, which can be expressed either by linguistic terms or in terms of relationships between numerical values. Fuzzy classifications of video sequences are also proposed in [6,7,11].

1.2 Contributions of This Paper

In this paper we propose a novel methodology to characterise produced multimedia content, based on fuzzy set theory [27] and multimodal content analysis. The central point of our theory is the concept of *Programme Unit*. The Programme Unit is a semantically closed entity identifiable during the fruition of a multimedia event. A Programme Unit either exists independently or is included into a more general one. For example, each topical part of a talk show is a Programme Unit, because it defines a semantically complete entity (thus completely understandable by the users) belonging to the main programme (thus contained into a programme at a higher level of representation).

First, we associate to each multimedia Programme Unit a *hierarchically structured surrogate*, i.e. a vector which is a combination of several sub-vectors, each representing a specific subview on the data (Section 2). Sub-vectors refer to specific properties characterising the objects at different levels. The surrogate contains data extracted by processing the audiovisual content, and takes into account visual, aural, and textual features.

Then, we present a learning system to cluster the surrogates of a given set of Programme Units. In particular, starting from seven commonly accepted classes, representing standard television genres, we rely on a statistical analysis technique to estimate the number of clusters in the domain for each selected feature subset of the surrogate (Section 3). Programme Units within the same cluster are likely to have similar characteristics in terms of (at least a subset of) their subview components. The resulting clusters might directly correspond to well known and accepted programme genres, as well as to *new interesting* concepts identified on the basis of some property correlation. For each Programme Unit in the learning set domain, we evaluate its (fuzzy) degree of membership to every identified cluster and create the Programme Unit fingerprints in terms of such membership degrees (Section 3). In Section 4 we show how these (novel) fingerprints can be used for similarity based information retrieval. Section 5 presents our empirical observations and, finally, Section 6 concludes the paper.

2 Programme Surrogates

Effective multimedia indexing and retrieval requires a multimodal approach, considering aural, visual, and textual information altogether. The surrogate used to characterise multimedia content is multimodal in that it *represents each Programme Unit as a set of extracted properties.*

Definition 1 (Programme Unit representation.). *Every Programme Unit, u, is represented by its surrogate $SUR(u) = \langle id_u, \overline{P_u} \rangle$, where id_u is the Programme Unit identifier and $\overline{P_u} = [\overline{F}, \overline{S}, \overline{C}, \overline{A}]$ is the* characteristic vector*, which represents the properties extracted from the numerical processing of the audiovisual material associated with the Programme Unit u. Each subvector $\overline{F}$, $\overline{S}$,$\overline{C}$, and $\overline{A}$ carries specific information: $\overline{F}$ represents the features at the audiovisual low-level, $\overline{S}$ carries the structural features, $\overline{C}$ carries cognitive features part, and $\overline{A}$ contains the audiovisual terms features.*

Figure 1 illustrates the process that, given a Programme Unit u, returns its characteristic vector $\overline{P_u}$. In [14] the physical architecture designed and implemented to calculate the Programme Units surrogates is presented.

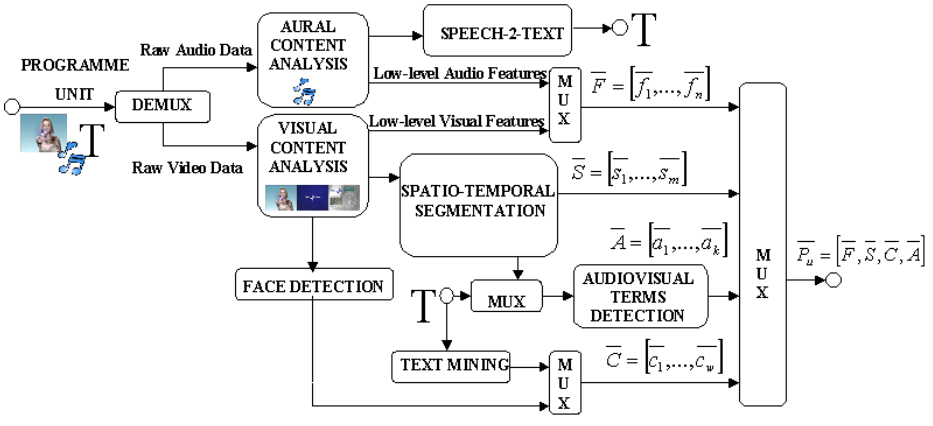


Fig. 1. Architecture for the multimodal feature extraction

2.1 The Low-Level Audiovisual Features Vector Component

Let $SUR(u) = \langle id_u, \overline{P_u} \rangle$, and let $\overline{F_u}$ denote the **audiovisual subcomponent** of $\overline{P_u}$. $\overline{F_u}$ is a vector of real numbers, which can in turn be grouped in a number of sub-vectors ($\overline{F_u} = [\overline{f_1}, \overline{f_2}, \ldots, \overline{f_n}]$), each representing a specific content-related feature extracted from the audiovisual material associated to the Programme Unit. Colour histograms and texture description are examples of such features.

In our current implementation, the low-level visual descriptors include the HSV colour features, the luminance Y, two Tamura texture features [22]- contrast (a measure of luminance difference between contiguous areas, [1]) and directionality (a measure of the main direction of the objects depicted in the frame[12])- and temporal activity information (computed by the luminance Displaced Frame Difference), for window size $t = 1$. Each feature is expressed by a 65-bins histogram, in which the last bin collects the number of pixels for which the feature is undetermined. The low-level audiovisual features are natively computed on a frame by frame basis (e.g. there is one hue histogram for each frame). As a global characterisation valid for the whole Programme Unit we use cumulative distributions of features over the number of frames of the Programme Unit.

2.2 The Structural Features Vector Component

The **structural features part** is constructed from the structural information extracted by the Segmentation Module. Structural properties include (i) the *total number of shot clusters*, (ii) the *shot clusters coverage properties* (which describe to which extent shot clusters cover the Programme Unit timeline), (iii) the *shot clusters configuration properties* (which characterise the clusters intrinsic features, such as elongation, total duration of included shots, cluster density - i.e. the amount of not included shots that are temporally internal to the cluster extension), (iv) the *shot clusters distribution properties* (which describe how the shot clusters are distributed along the Programme Unit timeline), and (v) the *shot clusters relationships properties*, which capture the mutual temporal relationships (such as adjacency, concatenation, inclusion) among clusters.

The structural properties of a Programme Unit u are represented by the vector $\overline{S_u} = [\overline{s_1}, \ldots, \overline{s_5}]$, in which each subvector $\overline{s_i}$ is associated with one of the five classes of structural properties described above.

2.3 The Audiovisual and Cognitive Vector Components

The **AudioVisual Terms Detection** module performs a statistical analysis to find out the frequency of some peculiar characteristics of audiovisual works carrying semantic information (for example the number of shots whose duration exceeds a predefined threshold, or the average number of words per second). The result is a vector $\overline{A}$ whose elements represent the frequency of the corresponding audiovisual term in the Programme Unit. Text mining and face detection techniques are used to extract *cognitive metadata* such as a frequency distribution of keywords from a pre-defined dictionary or the face-per-frame rate[1].

The subvectors of $SUR(u)$ characterising the cognitive features, audiovisual terms and the textual features have a structure similar to the low-level audiovisual and to the structural components.

3 The Programme Unit Fingerprints

A *fingerprint* is a representation that univocally characterises a given object. The concept has been recently adopted in many areas, from bioinformatics [13,18,19] to various multimedia applications [9,23,25]. We define our fingerprints as vectors in a fuzzy space. The *fingerprint construction* can be seen as a *space transformation* process, which maps each Programme Unit description from the surrogate's vector space to a new fuzzy vector space, whose dimensions are the number of clusters in the database, and whose elements are (fuzzy) degrees of membership of the Programme Unit to the corresponding cluster. Next, we present the steps of the fingerprint construction process.

[1] In the experimental prototype the face detection task is performed using the RT-FaceDetection library offered by Fraunhofer IIS.
http://www.iis.fraunhofer.de/bv/biometrie/tech/index.html

Clustering on the Training Set. For each sub-view of the surrogate, we *run a Fuzzy C-Means (FCM) clustering algorithm* [10] over the set of Programme Units in the training set. The value for C in the fuzzy clustering is determined empirically. For that purpose, we use the whole database as the data training set. The seven commonly considered television programme genres (Newscasts, Commercials, Cartoons, Football, Music, Weather Forecasts, and Talk Shows), identified manually on the data, are used as the starting clusters.

Let sv be any subview represented by the vector $\overline{x} \in \Re^n$, where n is the dimension of the surrogate related to sv (i.e. a $65 - bins$ histogram of the shot length distribution) and consider the set $MC = \{c_1, c_2, \ldots, c_k\}$ of the manually identified classes on the training set items. For every $i - th$ class $c_i \in MC$, and every sub-view sv, evaluate $\mu_\mathbf{i}$ and $\mathbf{\Sigma_i}$ (i.e. the average values vector and the covariance matrix restricted to the i-th class) and then calculate C as:

$$C = \sum_{i=1}^{k} \lceil \rho(\mathbf{\Sigma_i}, \mu_\mathbf{i}) \rceil \tag{1}$$

The function $\rho(\mathbf{\Sigma}, \mu)$ needs to have a saturating behaviour (towards C_{MAX}, which is the maximum number of clusters that we can have for each of the classes), in order to limit the maximum number of initial clusters for the fuzzy clustering algorithm. This way, sub-views that show higher variance/mean ratio (i.e. are more *spread*) over the initial classes contribute with a higher number of initial clusters, while those with lower ratio contribute with a lower number. In particular, we use the following function with these properties:

$$\rho(\mathbf{\Sigma}, \mu) = [1 + (C_{MAX} - 1)(1 - e^{-\frac{Tr(\mathbf{\Sigma})}{n \cdot \|\mu\|^2}})], \tag{2}$$

where $\rho(\mathbf{\Sigma}, \mu) \in (1, C_{MAX}]$ and $Tr(\mathbf{\Sigma}) = \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{\sigma_{i,i}^2}$. The initial positions of the clusters for the FCM are chosen randomly in the hyper-region expressed by:

$$a \cdot \sum_{i=1}^{n} \frac{(\mathbf{x_i} - \mu_\mathbf{i})^\mathbf{2}}{\sigma_{i,i}^2} \leq 1, \tag{3}$$

which denotes the ellipsoid having semi-axis lengths $l_i = 1/2$, $\forall i = 1, \ldots, n$ for $a = 4$ and $\sigma_{i,i} = 1$.

Example 1. Table 1 reports the mean and standard deviation values for the average shot length (ASL) distributions obtained for the Programme Units' of the learning set separated depending on the reference semantic classes in our multimedia database. Choosing $C_{MAX} = 5$ we obtain $C = 17$.

The choice of the most suitable and efficient distance metric to be used in the *FCM* process depends on the nature of the sub-views on which the clustering is applied. For features represented as histograms, a suitable distance metric is the histogram intersection, while for count vectors (as those of the structural properties) the application of a similarity measurement downstream of a Principal Component Analysis (e.g. Singular Value Decomposition) is applicable.

Table 1. Mean and Standard Deviation (SD) values for the average shot length (ASL) distributions obtained for each of the semantic classes in the database

	News	Commercials	Cartoons	Football	Music	W. F.	Talk Shows
ASL Mean [s]	4.59	1.43	1.6	5.03	3.93	11.79	10.4
ASL SD [s]	4.02	0.36	0.52	1.49	7.08	6.26	4.87

Programme Units Clustering. The fuzzy C-means algorithm is run on the set of Programme Units of the learning data set (the entire database), restricted to a specific sub-view of each of the four views in the features vector. For each Programme Unit, we define an *associated vector of fuzzy membership degrees* (one for each cluster) to contribute to its fingerprint w.r.t. that subview. The number of clusters (and the size of the fingerprint) can be reduced, by a cluster filtering algorithm [28].

Fingerprint Construction. At this point, every surrogate's subview has been mapped on a vector whose dimension is the set of "relevant" clusters, and whose values are the degrees of membership to the corresponding cluster. *The set of these vectors for a given Programme Unit represents the fingerprint for that unit.* The cross-referencing of these vectors, visualised as the Cartesian product of the individual components of the fingerprint, provides an *extended* fingerprint.

Example 2. Let V be a view, consisting of two sub-views, SV_1 and SV_2 respectively, corresponding to the two features $\overline{f_1} = [c_{1,1}, c_{1,2}, c_{1,3}] \in \Re^3$ and $\overline{f_2} = [c_{2,1}, c_{2,2}] \in \Re^2$. Let u be a Programme Unit that, after the fuzzy clustering process over the entire learning set, has been mapped in the clustering space $\Phi = (SV_1 \times SV_2)$. The corresponding fragment of the "extended" fingerprint will have 6 (i.e., 3×2) elements, and will look like the matrix $\overline{\overline{F_p}} = [c_{1,1}c_{2,1}; c_{1,1}c_{2,2}; \ldots; c_{1,3}c_{2,2}] \in \Re^{6,2}$.

We note that the extended fingerprint is virtual; i.e., the Cartesian product of the fingerprint components is not materialised. We remark that the fingerprint representation has a number of advantages. First, the fingerprinted Programme Unit is still in vectorial form, thus, we can still rely on the vector model for data retrieval. At the same time, we can rely on the fuzzy logic theory to express the uncertainty in the choice of a concept/class associated to a Programme Unit. In addition, especially when we consider the extended representation, the fingerprint makes it possible to discover new semantic information associated to each particular feature combination of the multimedia content.

4 Use of the Fingerprint

The primary use of the fingerprints is the identification of clusters of Programme Units within the learning set that possibly show a semantic relevance. Each extended fingerprint can be seen as a matrix of real numbers $\Phi \in R^{S \times V}$, where

V is the total number of sub-views and S is the product of the cardinalities of the set of clusters associated to the individual subviews in the surrogate (in Example 2, $V = 2$ and $S = 6$). Therefore, *the Programme Units can be clustered on the basis of their fingerprints*, by applying a (fuzzy) clustering algorithm that uses a suitable intra-matrix distance metric.

To stress on the importance of situations with high row-similarity among the fingerprints, we propose the following distance metric (called Non Linear Row Count, NLRC): given $A, B \in R^{S \times V}$, $A = (r_{a1}, \ldots, r_{aS})$, $B = (r_{b1}, \ldots, r_{bS})$

$$d(A,B) = \Omega\left(\frac{f_{com}(A,B,\epsilon)}{S}\right), \qquad f_{com}(A,B,\epsilon) = \sum_{i=1}^{S} \delta_i \tag{4}$$

where δ_i is a binary function that is equal to 1 if $|r_{aij} - r_{bij}| < \epsilon$, $\forall j = 0, \ldots, V$.

$$\Omega(x) = \frac{e^{-\alpha x}}{1 - e^{-\alpha}} + \frac{1}{1 - e^{\alpha}} \qquad \text{with } \alpha > 0 \qquad \text{and } 0 \leq x \leq 1. \tag{5}$$

$\Omega(0) = 1$, $\Omega(1) = 0$, $\Omega'(x) < 0$. The parameter α governs the speed with which the distance converges to zero with the variation of the number of common rows measured by $f_{com}(A, B, \epsilon)$. This function sums up the number of similar rows between the two matrices, by considering them similar if, for each couple of corresponding elements, the absolute difference is lower than a threshold ϵ.

The proposed distance metric is used as the core distance calculation of a second FCM process, this time performed using the extended fingerprint matrices as the feature vectors instead of the individual sub-view vectors. This second level of clustering will partition of the learning set in several fuzzy classes.

5 Empirical Observations

5.1 Benefits in Using the Extended Fingerprint

Our framework has been experimentally tested in a simple classification task involving the identification of the classes to which each Programme Unit in the training set belongs.

In order to simplify the interpretation of the results, we used only the structural view, consisting of three (mono-dimensional) sub-views: average visual shot length (ASL), visual shot cluster saturation[2] (CLS), and visual shot cluster duration[3] (CLD).

[2] Visual shot cluster saturation is defined as the ratio between the number of visual shots aggregated in shot clusters of at least two elements and the total number of shots of a Programme Unit.

[3] Visual shot cluster duration is the ratio between the total duration of all the shots that are aggregated in shot clusters of at least two elements and the Programme Unit duration.

Table 2. Results of the classification based on concatenated and combined fingerprints

	Cart.	Comm.	Football	News	Talk Shows	W. F.	**memb. STD**
F-meas. Conc.	0.63	0.72	0.39	0.67	0.58	0.57	0.14
F-meas. Comb.	0.44	0.66	0.48	0.47	0.80	0.71	0.05

To observe the effect of concatenated versus independent treatment of the subviews, we employed two different strategies. First, we considered the *concatenation* of the three subviews as a single (three-dimensional) feature vector, and we applied our fingerprinting process on it over the learning set database. Since all three features are considered together during clustering, this strategy does not lead into Cartesian product-based extension. The second strategy involves the application of the fingerprinting process independently on each of the three sub-views and the *combination* of these into an extended fingerprint.

In the experiment, the fuzzy clustering method had C set to 7. The parameters α and ϵ for the NLRC distance computation were set to 10 and 0.02 respectively. The fingerprints from the previous phase were classified into six genre classes using simple monodimensional linear classifiers. Table 2 shows the performances of the best classification results for both fingerprinting schemes.

In some cases the combined fingerprint shows better results than the concatenated one, while in other cases the reverse holds. Our interpretation is that in cases in which few feature configurations are dominant in explaining a class (e.g. commercials and cartoons have typically low ASL and CLD) the combined fingerprint performs worse. In turn, in classes in which the possible structural configurations are less regular (e.g. Talk-shows, Weather Forecasts and Football) the performance boosting is important, enhancing situations in which fingerprint elements combinations unveal semantic aggregation power that the concatenated approach does not succeed in evidentiating.

The experiments also showed that the average normalised membership to each of the (7) considered clusters using NLRC distance retained a standard deviation significantly lower than the standard deviation obtained with the Euclidean distance chosen for the concatenated case. This indicates a greater discrimination power (i.e. clusters are more dense).

5.2 Latent Semantic Characterisation Using Class Projection

As a second experiment, we used a test set of about 60 Programme Units that were not part of the learning set. These were not pre-annotated. The objective of the experiment was to obtain a projection of the test Programme Units on six genre classes and to analyse the resulting aggregations from the structural point of view. This task is different from the traditional classification task, in which test units are crisply classified towards the learnt classes. In this case we characterise test units fuzzily, in terms of the knowledge wired in the system during the training phase. For each of the Program Units we calculated a projection vector as $p = \overline{\overline{F(m)}} \times \overline{m}$.

Table 3. Classification clusters for the projection data

Cluster id	Cartoons	Commercials	Football	News	Talkshows	Weather forecasts
1	0.08	0.06	0.04	0.05	0.06	0.04
2	0.07	0.07	0.29	0.09	0.06	0.07
3	0.24	0.35	0.04	0.08	0.02	0.03
4	0.02	0.04	0.00	0.00	0.05	0.68

$\overline{m}$ is a 7×1 vector containing the fuzzy degrees of membership to the seven clusters ($C = 7$). The matrix $\overline{\overline{F(m)}}$ represents the cluster representativeness of the six genre classes. In fact each element f_{ij} of the matrix is the empirical F-measure with which the cluster j explains the class i. The vector p resulting from the multiplication of the 6×7 matrix $\overline{\overline{F(m)}}$ and 7×1 vector $\overline{m}$ is a 6×1 vector in which each element is the fuzzy degree of membership of the given programme unit to the corresponding class.

We then performed a clustering process on the resulting set of vectors p for the test Programme Units, finding out 4 meaningful clusters, whose centroids are reported in Table 3. Thus, the test units were projected to 4 cases with respect to their structural similarity in terms of the used features (ASL, CLD, CLS): one of these expressing no particular similarity to any of the classes, one expressing weak similarity to Football, one weakly similar to both Cartoons and Commercials, one definitely Weather Forecasts.

6 Conclusions and Future Work

In this paper we have introduced a new approach to multimedia data (and in particular television programmes) fingerprinting, based on the application of multimodal content analysis and fuzzy clustering. Our fuzzy fingerprinting has a number of important features:

(i) It preserves the structural information associated to Programme Units' surrogates. Whenever a new surrogate is defined as a vector (or a set) of subvectors associated to possibly independent concepts, the corresponding fingerprint will be another vector (or set), whose elements are in one to one correspondence to the ones of the surrogate. This representation defines a new information space, in which each multimedia document is represented in a semantic hierarchical structure.

(ii) The fuzzy fingerprint model is well suited for similarity based information retrieval, and it captures "semantic" similarities coming from common pattern in the programme data, at different semantic levels. In fact the modular structure of the fingerprint and its structural preservation property, make it possible to restrict the comparisons required by the retrieval process to subvectors of the fingerprint, thus investigating different possible aggregations of objects showing semantic relevance rather than assuming a static association between Programme Units and classification concepts.

(iii) Fuzzy fingerprints are the basis for a further classification step. The proposed architecture is based on the subview-based structured fingerprints and on the use of class projections using different contexts. It can be used to explore the construction of a fuzzy classification system in which users select the relevance of the available sub-views in their queries and the context against which the semantic mining of the tested element has to be evaluated.

Based on our empirical observations, we imagine a scenario in which the learning set is manually annotated with several classification schemes and a set of external entities have to be assessed with respect to one or more of these schemes. For example a scheme could be expressing affective annotations of programme units in natural language (e.g., using the terms "moving", "amusing", "intriguing"). The fuzzy projection set could offer to the user a hint of the possible emotions derived from the programme unit. If several annotations are present at the same time for the programme units in the learning set, they can be used as pluggable contexts for the projection of a test item. A major challenge we are currently addressing is the choice of the most promising combinations of features which can lead us to extract meaningful semantic information.

References

1. Battiato, S., Gallo, G., Nicotra, S.: Perceptive Visual Texture Classification and Retrieval. In: ICIAP 2003 (2003)
2. Bertini, M., Del Bimbo, A., Torniai, C.: Enhanced Ontologies for Video Annotation and Retrieval. In: ACM MIR (Multimedia Information Retrieval) Workshop (2005)
3. Dasiopoulou, S., Papastathis, V.K., Mezaris, V., Kompatsiaris, I., Strintzis, M.G.: An Ontology Framework For Knowledge-Assisted Semantic Video Analysis and Annotation. In: Proc. SemAnnot 2004 (2004)
4. Dimitrova, N., Agnihotri, L., Wei, G.: Video classification based on HMM using text and faces. In: Proc. European Signal Processing Conference (2000)
5. Dorado, A., Calic, J., Izquierdo, E.: A rule-based video annotation system. IEEE Transactions on Circuits and Systems for Video Technology 14(5), 622–633 (2004)
6. Doulamis, A., Avrithis, Y., Doulamis, N., Kollias, S.: Interactive Content-Based Retrieval in Video Databases Using Fuzzy Classification and Relevance Feedback. In: Proc. IEEE ICMSC 1999 (1999)
7. Ferman, A.M., Tekalp, A.M.: A fuzzy framework for unsupervised video content characterization and shot classification. J. of Electronic Imaging 10(4), 917–929 (2001)
8. Fischer, S., Lienhart, R., Effelsberg, W.: Automatic recognition of film genres. In: Proc. ACM Multimedia (1995)
9. Guerrini, F., Leonardi, R., Migliorati, P., Benini, S.: Effective Image Fingerprint Extraction Based on Random Bubble Sampling. In: WIAMIS 2004 (2004)
10. Höppner, F., Klawonn, F., Kruse, K., Runkler, T.: Fuzzy Cluster Analysis. Wiley, Chichester (1999)
11. Jadon, R.S., Chaudhury, S., Biswas, K.K.: Generic Video Classification: An Evolutionary Learning based Fuzzy Theoretic Approach. In: Proc. of the ICVGIP 2002 (2002)
12. Long, F., Zhang, H.J., Feng, D.D.: Fundamentals of Content-Based Image Retrieval. In: Multimedia Information Retrieval and Management- Technological Fundamentals and Applications. Springer, Heidelberg (2003)

13. Mezei, M.: A novel fingerprint for the characterization of protein folds. Protein Eng. 16(10), 713–715 (2003)
14. Montagnuolo, M., Messina, A.: Multimedia Knowledge Representation for Automatic Annotation of Broadcast TV Archives. In: Proceedings of the 4th Special Workshop on Multimedia Semantics (WMS 2006), pp. 80–94 (June 2006)
15. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
16. Rabiner, L., Juang, B.: An introduction to hidden Markov models. IEEE ASSP Magazine, 4–16 (1986)
17. Seymore, K., McCallum, A., Rosenfeld, R.: Learning hidden markov model structure for information extraction. In: AAAI 1999 Workshop on Machine Learning for Information Extraction (1999)
18. Sharan, R., Elkon, R., Shamir, R.: Cluster analysis and its applications to gene expression data. In: Ernst Schering workshop on Bioinformatics and Genome Analysis (2001)
19. Sharan, R., Maron-Katz, A., Shamir, R.: Click and expander:A system for clustering and visualizing gene expression data. Bioinformatics 19(14), 1787–1799 (2003)
20. Snoek, C.G., Worring, M.: Multimodal video indexing: A review of the state-of-the-art. In: Proc. Multimedia Tools and Applications (2003)
21. Takagi, S., Hattori, S., Yokoyama, Y., Kodate, A., Tominaga, H.: Sports video categorizing method using camera motion parameters. In: Proc. of the Visual Communications and Image Processing 2003 (2003)
22. Tamura, H., Mori, S., Yamawaki, T.: Texture features corresponding to visual perception. IEEE Trans. on Systems Man Cybernet 8(6), 460–473 (1978)
23. Tang, S., Li, J.T., Zhang, Y.D.: Compact and Robust Fingerprints Using DCT Coefficients of Key Blocks. LNCS. Springer, Heidelberg (2005)
24. Truong, B.T., Dorai, C.: Automatic genre identification for content-based video-categorization. In: Proc. Int. ICPR 2000, Barcelona, Spain (2000)
25. Venkatachalam, V., Cazzanti, L., Dhillon, N., Wells, M.: Automatic identification of sound recordings. IEEE Signal Processing Magazine 21(2), 92–99 (2004)
26. Xu, L.Q., Li, Y.: Video classification using spatial-temporal features and PCA, Proc. IEEE Inter. In: Conf. on Multimedia and Expo (ICME 2003) (2003)
27. Zadeh, L.A.: Fuzzy Sets as a Basis for a Theory of Probability. Fuzzy Sets and Systems (1978)
28. Pal, N.R., Bezdek, J.C.: On cluster validity for the fuzzy c-means model. IEEE Trans. on Fuzzy Systems 3(3), 370–379 (1995)

VeXQuery: An XQuery Extension for MPEG-7 Vector-Based Feature Query

Ling Xue[1], Chao Li[2], Yu Wu[1], and Zhang Xiong[2]

School of Computer Science and Technology, Beihang University
No. 37 Xue Yuan Road, Haidian District, Beijing, P.R.China
{xueling,wuyu}@cse.buaa.edu.cn,{licc,xiongz}@buaa.edu.cn

Abstract. MPEG-7 (Multimedia Content Description Interface) is a standard for describing the content of multimedia data and adopts XML (eXtensible Markup Language) Schema as its Description Definition Language. However, as a forthcoming standard for querying XML data/documents, XQuery (XML Query Language) has weakness in the query of vector-based feature described in MPEG-7. Thus, we propose an extension to XQuery, that is, VeXQuery (Vector-based XQuery) language, to resolve the problem of vector-based feature query in MPEG-7 retrieval. To fulfill the vector-based feature query, VeXQuery gives a set of vector similarity measurement expressions and defines the formal semantics of VeXQuery in terms of XQuery itself. Through this approach VeXQuery can be integrated seamlessly into XQuery.

Keywords: XQuery, MPEG-7, content-based retrieval, feature-based query.

1 Introduction

As a standard for describing the content of multimedia data, MPEG-7 adopts XML Schema as its Description Definition Language (DDL). Such a standard description framework offers a means to attach metadata to multimedia content. Furthermore, through this approach we can transform the multimedia retrieval into the MPEG-7 retrieval, which considerably improves the accessibility to multimedia data. However, it is now essential but not sufficient to make multimedia data searchable [1],[2].

XQuery is a forthcoming standard designed by W3C for querying and formatting XML data [3]. Now it is winding its way toward recommendation. Since MPEG-7 documents are XML-based, XQuery seems to fits naturally as a language for querying MPEG-7 documents. As we know, much multimedia retrieval is implemented through the similarity measurement of multimedia features and these features are stored in the vector-based data type. Nevertheless, the vector-based data type is not a built-in type of XML Schema and XQuery does not provide relative operators or functions, which include the similarity measurement and the measurement results scoring and ranking. Besides, many other retrieval standards, such as SQL/XML, XIRQL, MMDOC-QL and etc also face the similar problem [4],[5],[6].

In this circumstance, to fit MPEG-7 documents retrieval on dealing with vector-based features, it is necessary to extend XQuery. Therefore, we design the VeXQuery,

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 34–43, 2009.

a language extension to XQuery. It gives a set of vector similarity measurement expressions. The notion of ranked results, which indicates sorting the query result according to the similarities, is also introduced in order to support threshold and top-K queries. Moreover, VeXQuery can be integrated seamlessly into XQuery via defining the formal semantics in terms of XQuery itself.

The rest of this paper is organized as follows. Section 2 introduces the limitations of XQuery in vector-based feature retrieval. Section 3 presents the design goals of VeXQuery and the limitations of the function-based approach in the language design. Section 4 gives the details of the VeXQuery expressions. Section 5 proposes the formal semantics of VeXQuery in terms of XQuery itself. Section 6 gives the details of the VeXQuery Search Engine. Finally, some conclusions are drawn in section 7.

2 The Limitations of XQuery in MPEG-7 Vector-Based Feature Retrieval

2.1 MPEG-7 Descriptions for Multimedia Content

MPEG-7 is an ISO/IEC standard developed by MPEG (Moving Picture Experts Group). The goal of the MPEG-7 standard is to provide a rich set of standardized tools to describe multimedia and in particular audio-visual content [7],[8],[9].To give an example of MPEG-7 description, we construct the mpeg7video.xml, which is a MPEG-7 document for the movie video named Crazy Rock. A clip of the mpeg7video.xml document is listed in the following.

```
<MPEG7 name="CrazyRockVideo" version="1.0">
  <ContentDescription xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType" >
      <Video id="CrazyRock">
        ……
        <Segment id="0001">
          <GoFGoPHistogramD Histogram-
TypeInfo="Average">
            <Histogram HistogramNormFactor="1000">
              <HistogramValues>
                72 2 2 40 3 0 ... 0 4 6 2 13 21 105
              </HistogramValues>
            </Histogram>
          </GoFGoPHistogramD>
        </Segment>
        ……
      </Video>
    </MultimediaContent>
  </ContentDescription>
</MPEG7>
```

From the above document, we can see that the histogram is defined as follows [8],[9]. The tag "HistogramValues" indicates histogram values of the bins in the histogram, which is represented by an integer vector and the tag "HistogramNormFactor" represents the normalization factor of the histogram. As we know, to implement

video query based on the histogram values, we should measure the vector-based similarity on the feature of the average histogram, which is indicated using the tag "GoF-GoPHistogramD".

2.2 Vector-Based Feature Query in MPEG-7

In general, the multimedia content query approaches can be approximately divided into two categories: semantic query and feature-based query. The former refers to query based on high-level semantics like person, location, motion and etc, which is close to users' comprehension. Differently, the latter approaches such as query-by-feature-value, query-by-example and query-by-sketch [1], refers to techniques focused on the low-level features like color, texture, spectrum and etc. In addition, the vector-based feature query is widely used in both categories above, such as motion, color, texture and etc. However, unfortunately, XQuery is inadequate for vector-based feature query. To explain this problem more concretely, we consider the following examples of video retrievals that aim at the mpeg7video.xml mentioned in section 2.1.

Threshold query: Find segment(s) from the *mpeg7video.xml* according to the similarity of the average histogram feature between the segment and the sample clip. If the similarity is less than a threshold T, return the id of the segment. In addition, order the satisfied segments by the similarities ascending. To achieve these query goals, we should introduce the expressions for similarity measurement into XQuery.

Top-K query: In video retrieval, users are often interested only in the top few results. Thus the language for MPEG-7 retrieval should provide methods for specifying this explicitly and enable the users to obtain the top-k results based on the similarities.

Composite query: When issuing the video retrieval, users may intend to specify a composite search condition in which different search condition can be specified with different importance. Under this condition, the importance of different features may be different. Thus users should be able to specify the importance of each feature in some way (e.g., using weights).

3 Design Principles and Alternative Approaches

As mentioned above, VeXQuery is a language extension to XQuery and offers both the end users and application programmers a set of expressions for vector-based feature query in MPEG-7 documents. For the end users, VeXQuery should provide a query language that is relatively easy-to-use. For the application programmers, VeXQuery should present a useful tool which can facilitate and speed-up the development of multimedia retrieval applications [1]. Considering these issues, we attempt to motivate and describe a set of design principles that satisfy any vector-based feature query extension to XQuery. Then we show why an alternative approach based on extensions to the XQuery functions fails to satisfy the proposed design principles due to the fundamental limitations of this kind of approach. This motivates the need for a more powerful approach, that is, VeXQuery, which we will discuss in the next section further.

3.1 Design Principles

In order to seamlessly integrate VeXQuery into XQuery, several design principles that the extension should follow are given here by referring to a FullText Search Extension to XQuery [10]. They are listed in the following.

DP1: Users should be able to specify the search context over which the vector-based feature query is to be performed. For instance, in the Threshold Query mentioned in section 2.2, the search context is limited to all the segments, and even within the segments, it is limited in the tag "HistogramValues".

DP2: Users should be able to specify the return context, or the part of the document collection to be returned. For instance, in the Threshold Query given in section 2.2, the return context is limited to the segment id.

DP3: Users should be able to express all kinds of operations for the vector-based similarity measurement. Some common measures such as the Euclidean measure and the Canberra measure can be used to obtain the similarities.

DP4: Users should be able to control how the similarity is calculated. For instance, in the Composite Query of section 2.2, users should be able to specify the importance of each feature in some way (e.g., using weights).

DP5: Users should be able to obtain the top-k results based on the similarities.

DP6: Users should be able to embed VeXQuery primitives into XQuery expressions and vice versa. This feature will enable users to do both similarity queries (using VeXQuery) and structured data queries (using XQuery).

DP7: There should be no change to the XQuery data model, for XQuery expressions are already fully compositional. Changing this data model will lead to changes of the existing definition of every XQuery expression.

DP8: VeXQuery should be extensible with respect to new similarity measure expressions according to new user requirements.

DP9: It should be possible to statically verify whether a query is syntactically correct or not. In other words, syntax errors should be able to be detected at compile time, which is crucial for building robust applications.

DP10: VeXQuery should allow for static type checking, which is important for the interpretation of query results. Furthermore, static type checking is already a built-in feature of XQuery and should be preserved in VeXQuery.

3.2 Limitations of Function Approaches

As we know, XQuery supports user-defined functions and we have considered extending XQuery through the function-based approach to resolve the MPEG-7 retrieval. However, the function-based approach has some limitations in satisfying the design principles given above.

First consider the following query. Find segments from the video according to the results of the similarity measurement on two different vector-based features. Here we use the weights to specify the importance of each feature. If the similarity of the two features between the example clip and a segment is less than a threshold T, return the id of this segment. The similarity based on the two features is calculated as follows. First the similarity of each feature is calculated and normalized respectively. And then the similarities of the two features are summed up with weights to calculate the final similarity. Using the function-based approach, the above query is as follows.

```
WeightedDistance(Weight*, Distance*)
```

where Weight* is a sequence of weights, which is a float list, and Distance* is a sequence of the similarities on different features, which is a double list. The return result of the WeightedDistance function is a double value. Here, each Distance in the sequence is the return result of a Distance function as follows, which is based on the *mpeg7video.xml* in section 2.1.

```
Dis-
tance("Euclidean",//Segment/GoFGoPHistogramD/Histogram/
HistogramValues,(20,14,…,26))
```

where the parameter "Euclidean" is an operation code for the similarity measurement. The parameter //Segment/GoFGoPHistogramD/Histogram/HistogramValues is an XQuery expression used to select the average histogram value of all the segments, which is a list of integer vectors. And the parameter (20,14,...,26) is the average histogram value of the sample clip, which is cast as an integer sequence.

Thus we can find that this function-based approach is not able to embed the Distance function into the parameter of the Distance* in the WeightedDistance function because a sequence of functions is not supported in this approach. Moreover, as we know, the number of the items in the Distance* and the Weight* should be equal in the calculation of the final similarity. However the number of items in Distance* and Weight* will not be checked until runtime. Thus this approach cannot verify whether the query is correct or not statically and this violates DP10. Besides, since the operation of the similarity measurement is represented as a parameter of the Distance function as a string of "Euclidean", its correctness cannot be check until runtime either and this violates DP9.

Therefore, the function-based language syntax has some fundamental limitations in meeting the design principles. Thus we propose the VeXQuery, an extension to XQuery. Because the syntax of VeXQuery fits within the framework of the XQuery, it can fulfill the design principles above.

4 VeXQuery Expressions

In this section, we will introduce the VeXQuery expressions, including the VDistanceExpr and VWeightDistanceExpr. VDistanceExpr provides an approach to measure the similarity of the vector-based features and VWeightDistanceExpr provides an approach to specify how the final similarity of several features is calculated in the composite query.

4.1 VDistanceExpr

The similarity measurement of the vector-based features can be represented by a quintuple (D, OP1, OP2, R, T), where D is the operation of similarity measurement such as Euclidean and Canberra, OP1 is one operand for the similarity measurement such as the average histogram indicated in the section 2.1, OP2 is the other operand, R is the comparison operator which is in the set of ('>', '<', '=', '<=', '>=', '<>'), and T is the threshold. Thus the Threshold Query in section 2.2 can be represented as a quintuple, that is, ('Euclidean', //Segment/GoFGoPHistogramD/Histogram/ HistogramValues, (20,14,...,26), '<', T). Thus the VDistanceExpr can be defined as follows (satisfying DP3, DP6) [11] .

```
VDistanceExpr := Expr "mi" | "man" |" eu "| "ca" |
                 "mah" Expr
```

where Expr is an XQuery expression indicating the OP of the quintuple. The operators of "mi", "man" and etc represent the D. The corresponding operations are listed in Table 1, which includes a set of distance measurements in common use.

Table 1. List of VeXQuery operators for the similarity measurement

Similairity Distance Operations	VeXQuery Operators
Minkowsky	mi
Manhattan	man
Euclidean	eu
Canberra	ca
Mahalanobis	mah

The VDistanceExpr returns a sequence of xs:double, which indicates the similarities of the vector-based feature between the segments and the sample clip. Thus the VDistanceExpr of the Threshold Query in section 2.2 can be expressed by

```
//Segment/GoFGoPHistogramD/Histogram/HistogramValues eu
(20,14,…,26)
```

This expression returns the Euclidean distances of the average histogram values between the segments in the mpeg7video.xml and the example clip. Here, the Expr //Segment/GoFGoPHistogramD/Histogram/HistogramValues is an XQuery expression used to select the average histogram values of all the segments, which is cast as a list of integer vectors. The Expr (20,14,...,26) is the average histogram value of the sample clip.

There are several key points can be seen in the above example. First, it shows how VDistanceExpr can limit the context, thereby satisfying DP1. In this example, the context is limited to the tag "HistogramValues". Second, since VDistanceExpr returns a sequence of xs:double , which is a build-in type of XQuery, it can be arbitrarily nested within other XQuery expressions, thereby satisfying DP6. Third, since VDistanceExpr returns a sequence of xs:double, it can be easily type-checked, thereby satisfying DP10.

4.2 VWeightDistanceExpr

As described above, the VDistanceExpr expression can express threshold query. However, it is not fit for the composite query. VeXQuery addresses this issue by providing the VWeightDistanceExpr expression, thereby satisfying DP4. Referring to the related previous work [11], we define VWeightDistanceExpr syntax by

```
VWeightDistanceExpr ::= Expr "vweightdistance"
                        VDistanceWithWeight
```

where VDistanceWithWeight specifies how the final similarity is calculated with added notion of weights. The syntax of the VDistanceWithWeight are as follows.

```
VDistanceWithWeight ::= VDistanceExpr "weight" Expr
                        |VDistanceWithWeight '&&' VDis-
                        tanceWithWeight "norm" Expr
```

The VWeightDistanceExpr expression also returns a sequence of xs:double, which indicates the final similarity calculated with added notion of weights in the composite query. An example of the composite query using the VWeightDistanceExpr expression is given below. It uses a weight of 0.6 for the average histogram and a weight of 0.4 for the edge histogram feature. Moreover, the distance of each feature is normalized by the factor α.

```
let $h := (20,14,…,26)
     $m := (27,12,…,25)
for $node in //Segment
  let $distance := vweightdistance
     $node/GoFGoPHistogramD/Histogram/HistogramValues
     eu $h weight 0.4
     && $node/EdgeHistogramD/Histogram/HistogramValues
     eu $m weight 0.6 norm α
where $distance < T
  return <Segment>
           <SegmentId>{$[node]/@id}<SegmentId>
           <eu>{$distance}</eu>
         </Segment>
```

where the $h and the $m represents the average histogram and the edge histogram feature of the sample clip respectively. Since the result of VWeightDistanceExpr is a sequence of xs:double, it can be easily type-checked, satisfying DP10 and be arbitrarily embedded in other XQuery expressions, thereby satisfying DP2. In particular, VWeightDistanceExpr can be used in conjunction with FLWOR to compute top-k search result, thereby satisfying DP5. The following example illustrates how to implement the top-k query in section 2.2.

```
let $h := (20,14,…,26)
  for $segment at $rank in
     for $node in //Segment
     let $distance :=
     $node/GoFGoPHistogramD/Histogram/HistogramValues
     eu $h
     order by $distance ascending
```

```
    return <Segment>
              <SegmentId>{$[node]/@id}<SegmentId>
              <eu>{$distance}</eu>
           </Segment>
where $rank<=K
return {$segment}
```

5 VeXQuery Semantics

In this section, we introduce the semantics of VeXQuery expressions. Before introducing the formal semantics of VeXQuery expressions, we first introduce the FeatureVector, a data type used in the definition of VexQuery Semantics.

5.1 FeatureVector

As described above, VeXQuery expressions are based on the vector data type. The vector data type defined in MPEG-7 includes integer, double and float vector. To define a data type on which all the VeXQuery expressions can operate, we model the FeatureVector as an XML element conforming to the following Schema [12].

```
<complexType name="FeatureVector">
  <choice>
    <element ref="IntegerVector"/>
    <element ref="DoubleVector"/>
    <element ref="FloatVector"/>
  </choice>
</complexType>
```

where the definitions of IntegerVector, DoubleVector and FloatVector are described in the MPEG-7 standard [8],[9].

5.2 Semantics of VDistanceExpr

As described in section 4.1, the VDistanceExpr expression provides an approach for expressing the operations of the similarity measurement on vector-based features. We now specify the formal semantics of VDistanceExpr using an XQuery function [12].

```
function vo:VDistance ($vector1 as vo:FeatureVector*,
             $vector2 as vo:FeatureVector*,
             $distanceType as xs:integer) as xs:double*
```

VDistanceExpr returns the similarity as xs:double*. The operation of similarity measurement is specified by the $distanceType as an integer, which can be added according to the users' requirements, thereby satisfying DP8.

5.3 Semantics of VWeightDistanceExpr

As described in section 4.2, The VWeightDistanceExpr expression specifies how the final similarity is calculated with added notion of weights. Its semantics is indicated as below [12].

```
function vo:VWeightDistanceExpr ($distance as
                        vo:VDistanceWithWeight*) as
xs:double*
```

where the vo:VDistanceWithWeight is defined as a sequence of value pair cast as xs:double, which is indicated as follows.

```
<xs:complexType name="DistanceWithWeight">
  <xs:sequence maxOccurs="unbounded">
    <xs:element name="Distance" type="double"/>
    <xs:element name="Weight" type="float"/>
  </xs:sequence>
  <xs:element name = "Factor" type= "integer">
</xs:complexType>
```

6 VeXQuery Engine

Figure 1 depicts the architecture of the VeXQuery implementation.

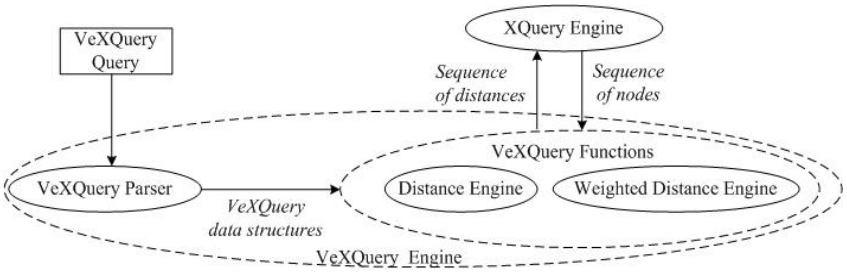


Fig. 1. The architecture of the VeXQuery search engine

The VeXQuery query proceeds as follows. When a VeXQuery query is entered, the VeXQuery engine parses the query and firstly identifies the VeXQuery expressions. Then translate the VeXQuery expressions into the XQuery functions based on the semantics described in section 5. Finally obtain the query results according to the results of similarity measurements.

7 Conclusion

This paper presents VeXQuery, a language extension to XQuery to resolve the problem of vector-based feature query in MPEG-7 retrieval. To realize the threshold query, the top-k query and the composite query exist in vector-based feature query, VeXQuery provides a set of expressions, including VDistanceExpr and VWeightDistanceExpr. VDistanceExpr provides an approach to measure the similarity of the vector-based features and VWeightDistanceExpr provides an approach to specify how the final similarity of several features is calculated. Moreover, the formal semantics of VeXQuery is also defined in terms of XQuery functions. Through this approach, VeXQuery can be integrated seamlessly into XQuery. And based on the language design, a VeXQuery engine is implemented.

References

1. Fatemi, N., Khaled, O.A., Cora, G.: An XQuery Adaptation for MPEG-7 Documents Retrieval. In: XML Conference & Exposition 2003. deepX Ltd., Dublin (2003)
2. Fatemi, N., Lalmas, M., Roelleke, T.: How to retrieve multimedia documents described by MPEG-7. In: van Rijsbergen, C.J., Ounis, I., Jose, J., Ding, Y. (eds.) Proceedings of the 2nd ACM SIGIR Semantic Web and Information Retrieval Workshop. ACM Press, New York (2004)
3. The World Wide Web Consortium. XQuery 1.0: An XML Query Language, http://www.w3.org/TR/2005/CR-xquery-20051103/
4. ISO/IEC JTC 1/SC 32 N1165, XML-Related Specifications (SQL/XML)
5. Fuhr, N., Groβjohann, K.: XIRQL: An extension of XQL for information retrieval. In: ACM SIGIR Workshop on XML and Information Retrieval. ACM Press, New York (2004)
6. Liu, P., Hsu, L.H.: Queries of Digital Content Descriptions in MPEG-7 and MPEG-21 XML documents. In: XML Europe 2002 Conference. deepX Ltd, Dublin (2002)
7. ISO/IEC JTC1/SC29/WG11N6828, MPEG-7 Overview
8. ISO/IEC JTC1/SC29/WG11/M6156, MPEG-7 Multimedia Description Schemes WD
9. ISO/IEC JTC1/SC29/WG11/M6155, MPEG-7 Multimedia Description Schemes XM
10. Yahia, S.A., Botev, C., Shanmugasundaram, J.: TeXQuery: A FullText Search Extension to XQuery. In: Feldman, S.I., Uretsky, M., Najork, M., Wills, C.E. (eds.) Proc. of the 13th Conf. on World Wide Web (WWW), vol. 594, pp. 583–594. ACM Press, New York (2004)
11. The World Wide Web Consortium. XQuery 1.0 and XPath 2.0 Functions and Operators, http://www.w3.org/TR/2005/CR-xpath-functions-20051103/
12. Amer-Yahia, S., Botev, C., Robie, J., Shanmugasundaram, J.: TeXQuery: A Full-Text Search Extension to XQuery Part I-Language Specification, http://www.cs.cornell.edu/database/TeXQuery/

Towards a Flexible Query Language

Rafik Bouaziz[1] and Salem Chakhar[2]

[1] Dep. of Computer Science, FSEG, University of Sfax,
Route de l'Aroport, BP 1088 3018 Sfax, Tunisia
raf.bouaziz@fsegs.rnu.tn
http://www.fsegs.rnu.tn
[2] LAMSADE, University of Paris Dauphine,
Place du Maréchal de Lattre de Tassigny,
75775 Paris Cedex 16, France
salem.chakhar@dauphine.fr
http://www.lamsade.dauphine.fr/~chakhar

Abstract. Defining and processing flexible queries is an important research topic in database area. We may distinguish two ways to support flexible querying in databases: (i) developing interface systems that allow queries in pseudo-natural languages, or (ii) developing an extended SQL-like languages. In this paper, we have adopted this second approach. The objective of this paper is to introduce a conceptual query language for accessing FSM-based databases. FSM is a data model that has been recently proposed.

Keywords: Fuzzy Database, Flexible Querying, Fuzzy Logic.

1 Introduction

Defining and processing flexible queries is one of the topics that seem to afford the greatest potential for further developments in database research area [2]. The key idea in flexible querying is to introduce preferences inside queries [2]. This means that the answers to these queries are no longer a crisp set and entities that "partially" much the preferences stated in the query are also provided to the user. We may distinguish two approaches to support flexible querying. The first approach consists in developing interface systems that allow queries in pseudo-natural languages (e.g. [8]). The second approach proposes SQL-like languages (e.g. [7]). In this paper, we have adopted this second approach. The general idea of this approach consists in introducing thresholds in the "FROM" and/or "WHERE" clauses to ensure that entities/tuples and/or attribute values verifying these threshold values are provided to the user. Since the attribute values may be fuzzy, conditions expressed in the "WHERE" clause are naturally fuzzy ones.

The objective of this paper is to introduce a conceptual query language for accessing FSM-based databases and illustrate some examples of data retrieve operations. FSM is a fuzzy semantic data model that has been recently proposed [3][4]. FSM is mapped to a fuzzy relational object database model (FRO) and

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 44–55, 2009.

implemented on PostgreSQL. The proposed query language uses the concepts of perspective class and qualification, introduced by [5], and introduces thresholds in the "FROM" and "WHERE" clauses. The thresholds in the "FROM" clause correspond to the *global degree of membership* (d.o.m) (cf. §2) and may be mapped to the support of fuzziness at entity/class level. The thresholds in the "WHERE clause correspond to the *partial* d.o.m. (cf. §2) and may be mapped to the support of fuzziness at the attribute level.

The paper is structured as follows. Section 2 briefly presents FSM. Section 3 addresses some implementation issues. Section 4 introduces the notions of perspective class and qualification, which are of importance in FSM query formulation. Section 5 deals with query formulation in FSM and provides some illustrative examples of data retrieve operations. Section 6 deals with query processing. Section 7 addresses some related work. Section 8 concludes the paper.

2 Fuzzy Semantic Model

In this section we briefly present FSM. We focalize only on concepts needed to introduce the proposed query language. Details can be found in [3][4]. The *space of entities* E is the set of all entities of the interest domain. A *fuzzy entity* e in E is a natural or artificial entity that one or several of its properties are fuzzy. In other words, a fuzzy entity verifies only (partially) some extent properties (see below) of its class. A *fuzzy class* K in E is a collection of fuzzy entities: $K = \{(e, \mu_K(e)) : e \in E \wedge \mu_K(e) > 0\}$. μ_K is a characteristic or *membership function* and $\mu_K(e)$ represents the *degree of membership* (d.o.m) of the fuzzy entity e in the fuzzy class K. Membership function μ_K maps the elements of E to the range $[0, 1]$ where 0 implies no-membership and 1 implies full membership. A value between 0 and 1 indicates the extent to which entity e can be considered as an element of fuzzy class K. A fuzzy class is a collection of fuzzy entities having some similar properties. Fuzziness is thus induced whenever an entity verifies only (partially) some of these properties. We denote by $X_K = \{p_1, p_2, \cdots, p_n\}$, $n \geq 1$, the set of these properties for a given fuzzy class K. X_K is called the *extent* of fuzzy class K. The extent properties may be derived from the attributes of the class and/or from common semantics. The degree to which each of the extent properties determines fuzzy class K is not the same. Indeed, there are some properties that are more discriminative than others. To ensure this, we associate to each extent property p_i a non-negative weight w_i reflecting its importance in deciding whether or not an entity e is a member of a given fuzzy class K. We also impose that $\sum_{i=1}^{n} w_i > 0$.

On the other hand, an entity may verify fully or partially the extent properties of a given fuzzy class. Let D^i be the basic domain of extent property p_i values and P^i is a subset of D^i, which represents the set of possible values of property p_i. The *partial membership function* of an extent property value is $\rho_{P^i_K}$ which maps elements of D^i into $[0, 1]$. For any attribute value $v_i \in D^i$, $\rho_{P^i_K}(v_i) = 0$ means that fuzzy entity e violates property p_i and $\rho_{P^i_K}(v_i) = 1$ means that this entity verifies fully the property. The number v_i is the value of the attribute

of entity e on which the property p_i is defined. For extent properties based on common semantics, v_i is a semantic phrase and the partial d.o.m $\rho_{P_K^i}(v_i)$ is supposed to be equal to 1 but the user may explicitly provide a value less than 1. More generally, the value of $\rho_{P_K^i}(v_i)$ represents the extent to which entity e verifies property p_i of fuzzy class K. Thus, the *global* d.o.m of the fuzzy entity e in the fuzzy class K is:

$$\mu_K(e) = \frac{\sum_{i=1}^{n} \rho_{P_K^i}(v_i) \cdot w_i}{\sum_{i=1}^{n} w_i}. \tag{1}$$

In FSM each fuzzy class is uniquely identified with a name. Each class has a list of characteristics or properties, called attributes. Some of these attributes are used to construct the extent set X_K defined above. To be a member of a fuzzy class K, a fuzzy entity e must verify (fully or partially) at least one of the extent properties, i.e., $\mu_K(e) > 0$. The classes in FSM are categorized as exact or fuzzy. An *exact class* K is a class that all its members have a d.o.m equal to 1. A *fuzzy class* K is a class that at least one of its members has a d.o.m strictly inferior to 1.

The elements of a fuzzy class are called *members*. In FSM, α-MEMBERS denotes for a given fuzzy class K the set $\{e : e \in K \wedge \mu_K(e) \geq \alpha\}$; where $\alpha \in [0, 1]$. It is easy to see that α-MEMBERS $\subseteq$ β-MEMBERS for all α and β in [0,1] and verifying $\alpha \geq \beta$. Note that 1-MEMBERS may also be refereed to *true* or *exact members*. In turn, α-MEMBERS with $0 < \alpha < 1$ are called *fuzzy members*.

FSM supports four different relationships: property, decision-rule, membering and interaction. The *property relationships* relate fuzzy classes to domain classes. Each property relationship creates an attribute. The *decision rule relationships* are an implementation of the extents of fuzzy classes, i.e., the set of properties-based rules used to assign fuzzy entities to fuzzy classes. The *membering relationships* relate fuzzy entities to fuzzy classes through the definition of their d.o.m. The *interaction relationships* relate members of one fuzzy class to other members of one or several fuzzy classes.

In FSM there are several complex fuzzy classes, that permit to implement the semantics of real-world among objects in terms of generalization, specialization, aggregation, grouping and composition relationships, which are commonly used in purely semantic modelling.

To close this section, we provide in Figure 1 a database example that illustrates most of FSM constructs. It will be used for illustration. In the example database, GALAXY is an aggregate fuzzy class whose members are unique collections of members from COMETS, STARS and PLANETS fuzzy grouping classes. These last ones are homogenous collections of members from strong fuzzy classes COMET, STAR and PLANET, respectively. NOVA and SUPERNOVA are two attribute-defined fuzzy subclasses of STAR basing on *type-of-star* attribute. PLANET-TYPES is an attribute-defined fuzzy composite class.

This composition is from PLANET fuzzy class basing on the *age* attribute. PERSON is an exact class. It has three enumerated subclasses: SCIENTIST,

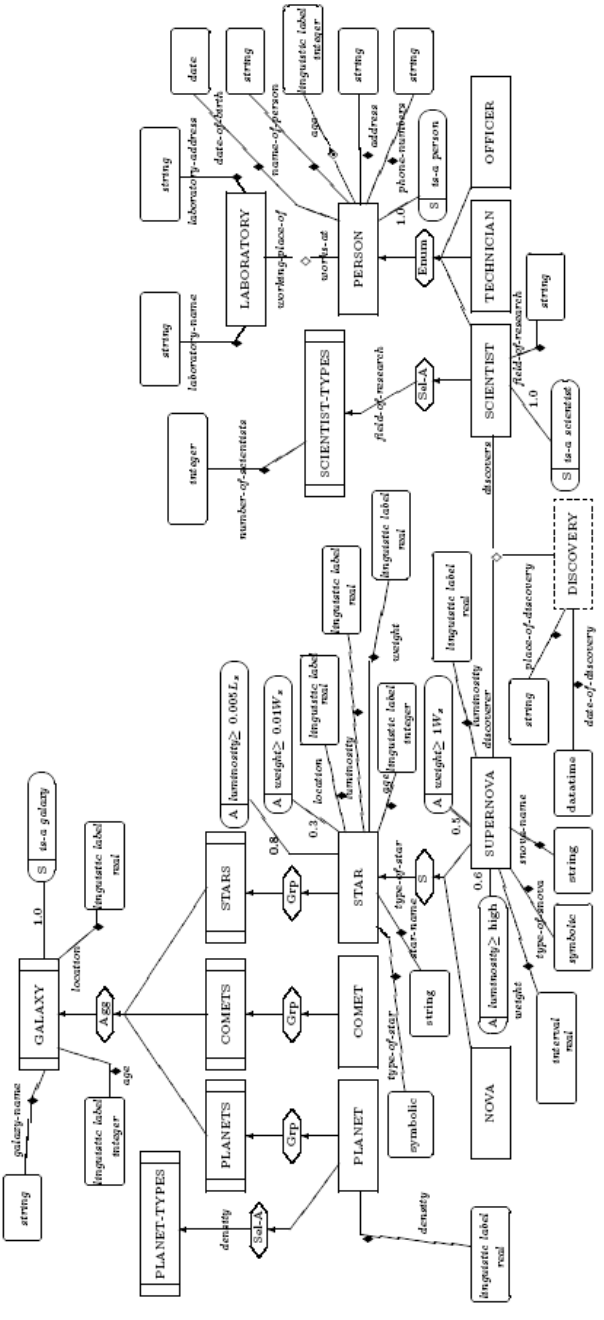


Fig. 1. Example of a FSM-based model

TECHNICIAN and OFFICER. Each person is affiliated with at least one LABORATORY. SCIENTIST is a collection of scientists and DISCOVERY is an interaction fuzzy class between SUPERNOVA and SCIENTIST. SCIENTIST-TYPES is a fuzzy composite class from SCIENTIST basing on *field-of-research* attribute.

3 Implementation Issues

3.1 Representing Imperfect Information

FRO supportes a rich set of imperfect data types (see [1] for details). First, it is important to mention that for facilitating data manipulation and for computing efficiency while giving the maximum flexibility to the users, the different types of attributes values in FRO are uniformly represented through possibility distribution. Each of these data types has one, two, three or four parameters permitting to generate its possibility distribution. For example, the graphical representation of possibility distribution of the *fuzzy range* data that handles the "more or less" information is provided in Figure 2. For instance, we may have: "*age* = more or less between 20 and 30". The d.o.m of any z in the fuzzy set A on which the attribute is defined is computed through Equation (2):

$$\mu_A(z) = \begin{cases} 1, & \texttt{if } \beta \leq z \leq \gamma; \\ \frac{\lambda - z}{\lambda - \gamma}, & \texttt{if } \gamma < z < \lambda; \\ \frac{z - \alpha}{\beta - \alpha}, & \texttt{if } \alpha < z < \beta; \\ 0, & \texttt{otherwise.} \end{cases} \tag{2}$$

The parameters β and γ represent the support of the fuzzy set associated with the attribute values and α and λ represent the limits of the transition zones.

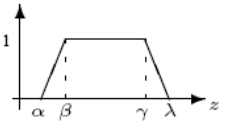


Fig. 2. Possibility distribution of "fuzzy range" data type

3.2 Implementing Imperfect Information

Several meta-relations have been defined to implement FRO. For example, to store the specificity of all the attributes, we define a meta-relation, called ATTRIBUTES with the following attributes: (i) *attr-id*: it uniquely identifies each attribute; (ii) *attr-name*: it stores the name of the attribute; (iii) *class-name*: denotes the fuzzy class to which the attribute belongs; and (iv)*data-type*: which is a multi-valued attribute that stores the attribute type. For crisp attributes, this attribute works as in conventional databases (it may take the values of integer,

float, etc.). For fuzzy attributes, the *data-type* attribute stores the fuzzy data type itself and the basic crisp data type on which the fuzzy data type is based. The ATTRIBUTES meta-relation associated with the model in Figure 1 is as follows:

attr-id	*attr-name*	*class-name*	*data-type*
attr-15	*star-name*	STAR	{string}
attr-20	*weight*	STAR	{interval, real}

At the extent definition of fuzzy classes, each attribute is mapped into a new composite with three component attributes: (i) *attr-value*: stores the value of the attribute as provided by the user; (ii) *data-type*: stores the data type of the value being inserted; and (iii) *parameters*: is a multi-valued attribute used to store parameters associated with the attribute value. The *data-type* attribute is used both at the extent definition and in the intent definition to allow users insert values of different data types, which may have different number of parameters. The mapping of the *weight* attribute of fuzzy class STAR is as follows:

weight
attr-value|*data-type*|*parameters*
{$10W_s$, real, {nil}}
{about $17W_s$, approximate value, {15,17,18}}

3.3 Mapping of a FSM-Based Model

As mentioned above, FSM-based model is mapped into a fuzzy relational object (FRO) database one. The FRO was implemented as a front-ends of the relational object database system PostgreSQL. Here we provide the transformation of only fuzzy subclass/superclass relationships. A fuzzy subclass B of a fuzzy superclass A is mapped into a relation which inherits all the attributes of the relation transformed from A. In addition to the attribute *dom*, the relation B contains a new attribute, denoted by *dom-A*, which is used to store the d.o.m of one entity from fuzzy subclass B in its fuzzy superclass A. The same reasoning is used for fuzzy subclasses with more than one fuzzy superclass. Note particulary that the relation mapped from fuzzy class B will contain several *dom-A*, one for each fuzzy superclass. For instance, the mapping of the fuzzy subclass SUPERNOVA in Figure 1 is as follows:

snova-name	*type-of-snova*	···	*dom*	*dom-star*
SN1987a	IIb	···	0.95	1.0
SN1006	Unknown	···	0.7	0.9

3.4 Computing the d.o.m

In FSM, an attribute-based extent property is associated with a condition of the form: <`left-hand-operand`> <`op`> <`right-hand-operand`>. The left-side parameter indicates the attribute name on which the rule is based. The right-side parameter may be a crisp or fuzzy value. The parameter *op* is a binary or a set

operator. For instance, we may have the following decision rules: *luminosity* $\geq$ `high`; and *age* $\in$ `[17-21]`. These operators may be associated with the negation operator "not". Basing on the work of [6], we have extended all the operators that may be used in the definition of the extent properties of fuzzy classes and for query processing. The extended operators apply both for crisp and imprecise data. In this second case, Zadeh's extension principle is used. For instance, the fuzzy "$\simeq$" that gives the degree in which two fuzzy numbers are approximately equal is computed as in Equation (3):

$$\mu_{\simeq}(\tilde{x}, \tilde{y}) = \begin{cases} 0, & \mid \tilde{x} - \tilde{y} \mid > \mathtt{m}; \\ 1 - \frac{\mid \tilde{x} - \tilde{y} \mid}{\mathtt{m}}, & \mid \tilde{x} - \tilde{y} \mid \leq \mathtt{m}. \end{cases} \tag{3}$$

The parameter `m` represents the common support of fuzzy numbers $\tilde{x}$ and $\tilde{y}$. The proposed calculus delivers a index of possibility and the computed degrees are values of possibility obtained through Zadeh's extension principle. A degree of necessity can be also computed.

4 Perspective Class and Qualification

In this section we introduce the concepts of perspective class and qualification. In the generic definitions below we adopt the following conventions: (i) [ ]: optional parameter(s); (ii){ }: list of parameters or values; (iii) | : the equivalent of the binary operator "xor"; (iv) < > : obligatory parameter(s); and (v) ( ) : series of parameters connected with the "xor" operator. The notion of *perspective class* is introduced in [5]. It is defined as the class with which the user is primarily interested when formulating his/her query. It simplifies query formation and allows users with different interests to approach the database from points of view appropriate to their needs [5]. The perspective class can be associated with an appropriate syntactic process, called *qualification*, allowing immediate attributes of other classes to be treated as if they were attributes of the perspective class. This process may be extended through the entity-valued attributes concept to the attributes related by more than one level of qualification. (The entity-valued attributes are specific, non printable, binary relationships that describe the properties of each entity of a class by relating it to an entity (or entities) of another (or the same) class.) These attributes are called *extended attributes*. For example, in Figure 1, *field-of-research* is an immediate attribute of SCIENTIST and *name-of-person* is an inherited attribute of SCIENTIST from PERSON.

Suppose that classes SCIENTIST and TECHNICIAN in Figure 1 are related and two entity-valued attributes *supervises* (from the point of view of SCIENTIST) and *supervisor* (from the point of view of TECHNICIAN) are defined for them. Then, with TECHNICIAN as perspective class, the *name-of-person* of *supervisor* refers to the name of a technician's supervisor(s) (i.e. a scientist entity). This last qualification is an extended attribute of TECHNICIAN in this example.

Furthermore, the notion of perspective class can be combined with generalization hierarchies to simplify query formation. For example, consider again the

hypothetical binary relationship between SCIENTIST and TECHNICIAN, then the list of technicians and the name of their supervisors can simply be obtained as follows (the syntax of a retrieve query in FSM is provided in §5):

FROM *technician* RETRIEVE *name-of-person*, *name-of-person* OF *supervisor*

In this example TECHNICIAN is the perspective class. This query lists the name of all technician and for each one it provides the name of his/her supervisor but if a technician has no supervisor, whose name should be returned with a null value for the supervisor's name attribute. In this example, the qualification avoided the necessity to put the SCIENTIST class in the FROM clause since the entities of this class are "reached" through the entity-valued relationship.

The general syntax of qualification of an attribute is as follows [5]:

<*attr-name*> {OF <*entity-valued-attribute-name*> [AS <*class-name*>]}
OF <*perspective-class-name*> [AS <*class-name*>]

The *attr-name* is either a data-valued or an entity-valued attribute. The "AS" clause specifies subclass/superclass role conversion (from a superclass to a subclass) in the same generalization hierarchy and may be best thought of as "looking down" a generalization hierarchy [5]. The following are some examples of qualification from Figure 1:

name-of-person OF *discoverer* OF *supernova*
laboratory-address OF *working-place-of* OF *person*

In the first qualification the perspective class is SUPERNOVA. It returns for each supernova the name(s) of its discoverer(s). The second one uses PERSON as the perspective class. It returns for each person in the database the address of the laboratories he works at.

5 Syntax of Retrieve Queries

The generic syntax of a retrieve query in FSM is as follows :

[FROM {(<*pers-class-name*> [WITH DOM $< op_1 >$ <*class-level*>] | $< \alpha -$MEMBERS OF *pers-class-name*>)}]
RETRIEVE <*target-list*>
[ORDER BY <*order-list*>]
[WHERE <*select-expression*> [WITH DOM $<$ op_2 $>$ <*attr-level*>]]

The argument of the FROM statement is a list of perspective classes names (*pers-class-name*) with their respective levels of selection (*class-level*) or a specification of the $\alpha-$MEMBERS to be considered. Only members that have a global d.o.m verifying the arithmetic comparison ensured by the operator op_1 (when the "WITH DOM" part is used) or have a d.o.m greater or equal to α are considered in the selection process. We remark that the WITH DOM part in the FROM clause is facultative and when omitted, all the entities of *perspective-class-name* that verify the WHERE clause are returned. This avoid the necessity of introducing the "WITH DOM > 0" condition when no restriction is imposed

on the global d.o.m. of the entities as in queries 4 hereafter. The *target-list* in the RETRIEVE statement is a list of expressions made up of constants, immediate, inherited and extended attributes of the perspective class, and aggregate and other functions applied on such attributes. The ORDER BY statement is used to choose the way the list of entities is ordered. The *select-expression* in the WHERE statement is a set of symbolic, numerical or logical conditions that should be verified by the attributes of all selected entities. When it is necessary, attributes-based conditions may be combined with appropriate selection levels (*attr-level*) and only entities that their attributes values have a partial d.o.m verifying the arithmetic comparison ensured by the operator op_2 are selected. The following are some illustrative examples of data retrieve operations taken from Figure 1.

Query 1. Retrieve the name and type of supernova that have global d.o.m equal to or greater than 0.7 and have luminosity greater than $15L_s$ with partial d.o.m equal to or greater than 0.9. The symbol L_s is the luminosity of the sun, often used as measurement unit.

```
FROM supernova WITH DOM ≥ 0.7 RETRIEVE snova-name, type-of-snova WHERE luminosity> 15L_s WITH
DOM ≥ 0.9
```

Query 2. Retrieve the name of all true supernovae and the names of their discoverers.

```
FROM 1-MEMBERS OF supernova RETRIEVE snova-name, name-of-person OF discoverer OF supernova
```

Here the qualification permits to avoid the necessity of adding the class SCIENTIST in the FROM clause. In addition, it avoids the necessity of a WHERE clause.

Query 3. Retrieve dates of discoveries and names of all supernovae of type "Ia" that are located in milky-way galaxy with a global d.o.m greater than 0.5 and having high luminosity with d.o.m less than 0.7.

```
FROM discovery, supernova, galaxy RETRIEVE snova-name, date-of-discovery
WHERE type-of-snova = "Ia" and (galaxy-name="milky-way" and galaxy.location = supernova.location
WITH DOM > 0.5) and luminosity= high WITH DOM < 0.7
```

In this query example as in the next one, several perspective classes are used. In addition, the "WITH DOM" part is omitted from the FROM clause and so the conditions of the WHERE clause will be checked for all the entities.

Query 4. Retrieve the name, the date of discovery and the discoverer of all supernovae which are not located in the milky-way galaxy with d.o.m not less than 0.5.

```
FROM supernova, discovery
RETRIEVE snova-name, date-of-discovery, name-of-person OF discoverer OF supernova
WHERE supernova.location not in (FROM galaxy RETRIEVE location WHERE galaxy-name="milky-way")
WITH DOM ≥ 0.5
```

This example illustrates an imbricated query in FSM.

6 Query Processing

The query processing schema (under implementation) contains three phases (see Figure 3):

- **Phase 1**: Syntactic analysis.
- **Phase 2**: Verification of the conditions specified in the `FROM` statement. It returns, for each tuple, a *global satisfaction degree* $d_g \in [0,1]$ measuring the level to which the tuple satisfies the *class-level conditions*. The tuples for which $d_p > 0$ represent the input for the next phase.
- **Phase 3**: Associate to each tuple a *partial satisfaction degree* $d_p \in [0,1]$ measuring the level to which tuples satisfy the *entity-level conditions*.

Then, the overall satisfaction $d_o \in [0,1]$ is computed as $d_o = d_g * d_p$.

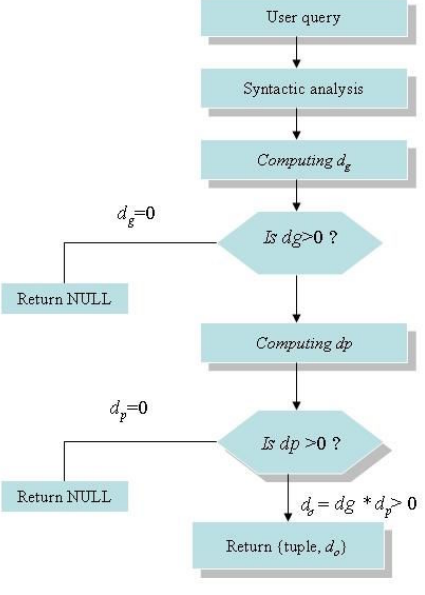


Fig. 3. Query processing schema

The conditions in the `WHERE` statement may be connected with `AND`, `A OR` or `NOT`. In this case, the computing of partial satisfaction degrees $d_p > 0$ are as follows (for two conditions c_1 and c_2):

Connector	$d_p(c_1)$	$d_p(c_2)$	d_p
AND	α	β	$\alpha * \beta$ or $\min\{\alpha, \beta\}$
OR	α	β	$\max\{\alpha, \beta\}$
NOT	α	–	$1 - \alpha$

where: $d_p(c_2)$ and $d_p(c_2)$ is the partial satisfaction degrees for c_1 and c_2, respectively; and α and β are in $[0,1]$.

7 Related Work

In [7], based on possibility distribution and semantic measure of fuzzy data, the authors first extend some basic concepts in object-oriented databases, including objects, classes, objects-classes relationships, subclass/superclass and multiple inheritances in order to support fuzzy information. The paper presents then a generic fuzzy object-oriented database model (FOODBM) and briefly introduces an extended SQL-like query langauge. The basic operations of projection, join and union under fuzzy information environment are also addressed in the paper.

The authors in [8] describe a flexible query interface based on fuzzy logic. The interface allow users to make questions in (quasi) natural langauge and obtain answers in the same style, without having to modify neither the structure of the database nor the DBMS query language. Hence, queries in natural langauge with pre-defined syntactical structures are performed, and the system uses a fuzzy natural language process to provide answers. This process uses the fuzzy translation rules of possibilistic relational fuzzy langauge proposed by [11]. One main advantage of this work is the fact that developers do not have to learn a new langauge, such as a new extension to SQL.

In [9], based on similarity relations, the IFO model was extended to the ExIFO (Extended IFO) to represent uncertainty as well as precise information. ExIFO support uncertainty at the attribute, entity, and instance/class levels. The authors provide also an algorithm for mapping the schema of the ExIFO model to an extended NF^2 database model. The paper includes also graphical representation and extended algebra operators such that projection, cartesian product and union, along with an extended SQL-like query language called XSQL.

The FOOD (Fuzzy Object-Oriented Database) model in [10] is a similarity-based fuzzy object-oriented data model. FOOD supports fuzziness at attribute, object/class as well as subclass/ supercalss relationships levels. In this paper, the authors propose a solution to the multiple inheritance problem. Equally, the paper introduces a prototype—which is implemented on the EXODUS storage manager system—and briefly presents the associated object definition language (ODL) and the object query langauge (OQL).

8 Conclusion and Future Work

In this paper, we have introduced an ongoing query language devoted to FSM-based databases. This query language uses the notions of perspective class and qualification. These concepts permits to simplify query formation and allows users with different interests to approach the database from points of view appropriate to their needs. When combined with fuzzy set theory, they provide a simplified and powerful query language.

The implementation of the proposed language is ongoing. The first step concerns the development of the lexical analyser and the parser. The role of the lexical analyser is to convert queries introduced by the user into a series of tokens that can be read by the parser. The role of the parser is to parse the queries

provided by the user into their equivalent algebraic form. Javacc was chosen as the implementation language.

Once the proposed query langauge is fully implemented, our attention will concerne the development and implementation of query optimisation strategies.

References

1. Bahri, A., Bouaziz, R., Chakhar, S., Naïja, Y.: Implementing imperfect information in fuzzy databases. In: Proc. SCIII 2005, Tunis, Tunisia, October 14-16 (2005)
2. Bosc, P., Kraft, D., Petry, F.: Fuzzy sets in database and information systems: Status and opportunities. Fuzzy Sets and Systems 156, 418–426 (2005)
3. Bouaziz, R., Chakhar, S., Mousseau, V., Ram, S., Telmoudi, T.: Database design and querying within the fuzzy semantic model. Information Sciences 177(21), 4598–4620 (2007)
4. Chakhar, S., Telmoudi, A.: Conceptual design and implementation of the fuzzy semantic model. In: Proc. IPMU 2006, Paris, France, July 2-7, pp. 2438–2445 (2006)
5. Fritchman, B.L., Guck, R.L., Jagannathan, D., Thompson, J.P., Tolbret, D.M.: SIM: Design and implementation of a semantic database system. In: ACM SIGMOD Conference, pp. 46–55 (1989)
6. Medina, J.M., Vila, M.A., Cubero, J.C., Pons, O.: Towards the implementation of a generalized fuzzy relational database model. Fuzzy Sets and Systems 75, 273–289 (1995)
7. Ma, Z.M., Zhang, W.J., Ma, W.Y.: Extending object-oriented databases for fuzzy information modeling. Information Systems 29, 421–435 (2004)
8. Ribeiro, R.A., Moreira, A.M.: Fuzzy query interface for a business database. International Journal of Human-Computer Stuies 58, 363–391 (2003)
9. Yazici, A., Buckles, B.P., Petry, F.E.: Handling complex and uncertain information in the ExIFO and NF^2 data models. IEEE Transactions on Fuzzy Systems 7, 659–676 (1999)
10. Yazici, A., George, R., Aksoy, D.: Design and implementation issues in fuzzy object-oriented data model. Information Sciences 108, 241–260 (1998)
11. Zadeh, L.A.: PRUF-a meaning representation language for natural languages. International Journal of Man-Machine Studies 10, 395–460 (1978)

Annotate, Query and Design Multimedia Documents by Metadata

Anis Jedidi[1], Ikram Amous[1], and Florence Sèdes[2]

[1] MIRACL, route de M'Harza, ISIMS, BP 1030, 3018 Sfax, Tunisie
anis.jedidi@isimsf.rnu.tn, ikram.amous@isecs.rnu.tn
[2] IRIT, 118 route de Narbonne, UPS, 31062 Toulouse, France
sedes@irit.fr

Abstract. In this paper, we focus on the managing of multimedia document and more precisely on the annotation and the generation of adaptable multimedia documents. Our solution is directed towards analysing the ways to "bridge the gap" between physical and semantic levels, for multimedia document modelling and querying. Our goal is to describe how to model and unify features elicited from content and structure mining. These descriptors are built from the various features elicited from the multimedia documents using available processing techniques. The personalization enables dynamic re-structuring and re-construction of hypermedia documents answering to the user queries. However, more factors should be considered in handling hypermedia documents. Once queried, documents can be adapted by using an indexing scheme, which exploits multiple structures. We can process queries efficiently with minimal storage overhead. We suggest for that, the adaptation of multimedia document content with user needs and preferences. This approach is based on the OOHDM methodology extension with the use of the metadata.

Keywords: metadata, annotation, spatiotemporal operators, querying, OOHDM Methodology, design.

1 Introduction

The multimedia document expansion involves a proliferation of information. However, the existing tools to organise, present and query these documents are still rather poor and remain to design and develop models and interfaces enabling an easy and sure handling of these heterogeneous resources. The multimedia documents are heterogeneous for several reasons: semantic heterogeneity of the contents, structural heterogeneity of the data and format heterogeneity (HTML, XML, MPEG-7...) of the documents.

Querying such resources is enabled by processes facilitating the access information independently of their heterogeneities. Elicitation and contents analysis tools can allow the identification of fundamental elements in the document structure. Thus, into the documentary information systems, a new concept called "metadata" gathering the descriptors is introduced to bring annotations to the multimedia documents. Generally, the metadata are added to the data to help, identify, describe and locate the various electronic resources.

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 56–66, 2009.

The metadata enabling annotation and description of multimedia documents may belong to two families: The metadata like histogram colours, size… which represent signal features and the metadata which represents semantic information easily expressed by the user. One of the last standards for the multimedia documents description is MPEG-7 standard. MPEG-7 can describe only created documents by the associated tools. Describing an already existing document with the descriptors of MPEG-7 standard requires a recreation of it.

The querying of multimedia documents, considered as semi-structured, is a major aim of the annotation of these documents by metadata. It is based on the contents and/or the document structure. The XQuery language is promoted by the W3C [5], but any other querying language could be used at the time.

Our contributions are at several levels, and try to propose descriptors, homogenise structures and enable the querying. The approach that we propose consists in specifying metadata families describing each type of medium. These metadata allow the "annotation" of document contents and its semantics. The proposal of a dynamic, flexible, evolutionary description enables the processing of legacies data and documents. Such a description does not take account the way in which the document was created nor by which tool it will be presented. The originality of our approach resides in its generality, since the documents are posterior analysed. Indeed, our annotation proposal relates to documents which are not standardised called legacy documents. Extracting metadata is based on a document indexing and segmentation process elaborated medium by medium.

Our metadata modelling proposal for each media type (text, image, audio, and video) is detailed in [3]. We propose the annotation of this media by a set of metadata. These metadata are structured in a meta-document having the root tag "MediaType_file". Each file is composed of documentary granules identified by a structure recognition process for each media type.

Once metadata are extracted, we propose to display adaptive multimedia document responding to the user's needs and queries. The multimedia document content contains all documentary units, extracted metadata and all that the user wants to see displayed. The adaptable multimedia documents are proposed to avoid the cognitive overload emerging to the user.

This paper is organized into five sections. In section two, we present our proposal for the multimedia document annotation by metadata. Section three develops our proposal for multimedia document querying. Section four presents our proposal for adaptable multimedia document by the extension of OOHDM methodology. Finally in section five, we present a validation of our proposal by a prototype.

2 The Meta-document Representation

Multimedia annotation is an extension of document annotation such as GDA (Global Document Annotation). A data model is a way of structuring data for access by a computer program. In this case, it is a way of storing the annotation on a structural graph. A data model should capture the important features of the real data and make them available to program in a natural way.

2.1 Annotation Graph Extension

An annotation graph [4] is a directed graph whose arcs represents structural relationships and temporal/spatial locations, adjacent to annotated regions and associated features.

Existing multimedia temporal annotation models can be divided into two classes: instant-based and interval-based models. In instant-based models, the elementary units are points in a time space. An example of these approaches is a timeline, in which media content is placed on several time axes called tracks, one for each media type. All events such as the start or the end of a segment are totally ordered on the timeline. This latter defines a set of element types that can be used with SGML document types to provide hyper linking and other facilities in a standardised way. HyTime neither compete with nor change SGML in any way. It is simply built on the basic facilities that SGML provides.

Interval based annotation models consider elementary media entities as time intervals ordered according to some relationships. Existing models are mainly based on the relations defined by Allen for expressing the knowledge about time [1]. The model is applied in SMIL.

The lack of content annotation issues in these related works led us to extend them to annotate multimedia documents by their content. Our proposal consists in a set of metadata, relative to the document content, modelled by XML. Indeed, once extracted, the metadata are tagged into documents called “meta-documents” [3]. Each meta-document references the "native" initial document it describes. Indeed, metadata must be helpful for users to access to the original documents, by keywords, size, speaker, image region, etc.

Identifying each item (unit, granule, object, etc.) is necessary for labelling and retrieving the result of a query. We chose a graphical representation to represent, describe and organise metadata. Eliciting and creating structure from each document allows the identification of document items called “elements”. An element can be a chapter, a section, a paragraph, image region, texture [6], sound key, speaker or video metadata identified by various segmentations and indexing techniques [11, 9].

Continuous media such as audio and video impose new requirements on document modelling. These requirements are essentially due to the intra-media and inter-media temporal and spatial information. Multimedia document architecture can be defined as an integrated and homogeneous collection of data describing, structuring the content and representing their temporal and spatial relationships in a single entity.

There are three ways of considering the structure of multimedia document: logical structure, spatial structure, and temporal one. Spatial structure usually represents the layout relationships and defines the space used for the presentation of an image. Temporal structure defines temporal dependencies between elements.

The spatial relationships have been widely studied as spatial operators for handling spatial objects. Several classifications have been proposed like [7]. Our purpose here is neither to compare them nor to define yet another one. So, we have chosen the classification proposed by [10], which groups all those spatial elementary operators into a few sets. The list of spatial operators in our proposal is: the disjunction (sd), the adjacency (sa), the overlapping (so), the inclusion (si).

There are two classifications of time relationships: The time between elements and the relationships between documents. The first class consists in intra-document relationships, referring to the time among various presentations of content elements. The second consists in the inter-document relationships, which are the set of relations between different documents (for example the synchronisation of two audio segments during a presentation). In our proposal, we develop the intra-documents' temporal links between the different elements.

2.2 Meta-document Annotation Graph

We propose to extend the meta-documents and the graphical representation by taking into account spatial and temporal operators in XML elements and attributes [3]. The spatial and/or temporal operators allow the exploitation of how two metadata are connected in space and/or in time. Considering the graphical representation of meta-document, the spatial and temporal operators are introduced as links in the graph. Let us consider a multimedia document containing audio and textual description. We present the graphical representation with regard to our DTD structure for text and audio documents. We respect also the additional structure for spatiotemporal relationship elicited by the annotation process of multimedia document. The complete graph contains links between metadata and identification of different elements is given Figure 1.

The corresponding meta-document is made of two parts. The first part -from <DOCUMENT…> to </DOCUMENT…>- describes the structural links between elements. The second part gathers spatial and temporal links between elements. These links are ordered according to their appearance into the initial document. This meta-document is as following:

```
<META_DOCUMENT>
<DOCUMENT id= #0 >
   <AUDIO_FILE id=#1.0>…</> <TEXT_FILE id=#1.1>……</>… …
</DOCUMENT>
<ST_LINKS>
   <TEMPORAL_LINK id1=#1.1    link="ts"    id2=#3.0/>
   <TEMPORAL_LINK id1=#3.0    link="tb(2.30)"    id2=#3.1/>
   <TEMPORAL_LINK id1=#3.1    link="tm"    id2=#2.5/>
   <TEMPORAL_LINK id1=#3.5    link="ts"    id2=#3.2/>
   <SPATIAL_LINK  id1=#4.3    link="sd-sw" id2=#4.2/>
   <SPATIAL_LINK  id1=#3.5    link="si"    id2=#2.5/>
</ST_LINKS>
</META_DOCUMENT>
```

One of the annotation advantages by XML structure is that it allows designing the semantic structure of a document explicitly. Our annotations are used to clarify document content for a human consumer. The key distinction of our annotation is that we annotate collections of multimedia document with the context of its composite presentation, rather than a single media or timeline of temporal axis. The spatial and temporal operators are introduced in this directive. These operators enable description on document and not its presentation like SMIL presentation.

SMIL 2.0 provides an attractive structure to support presentation requirements because of several key features. SMIL is an XML language. It is a declarative language, extensible which support extensive timing, layout, linking and content control

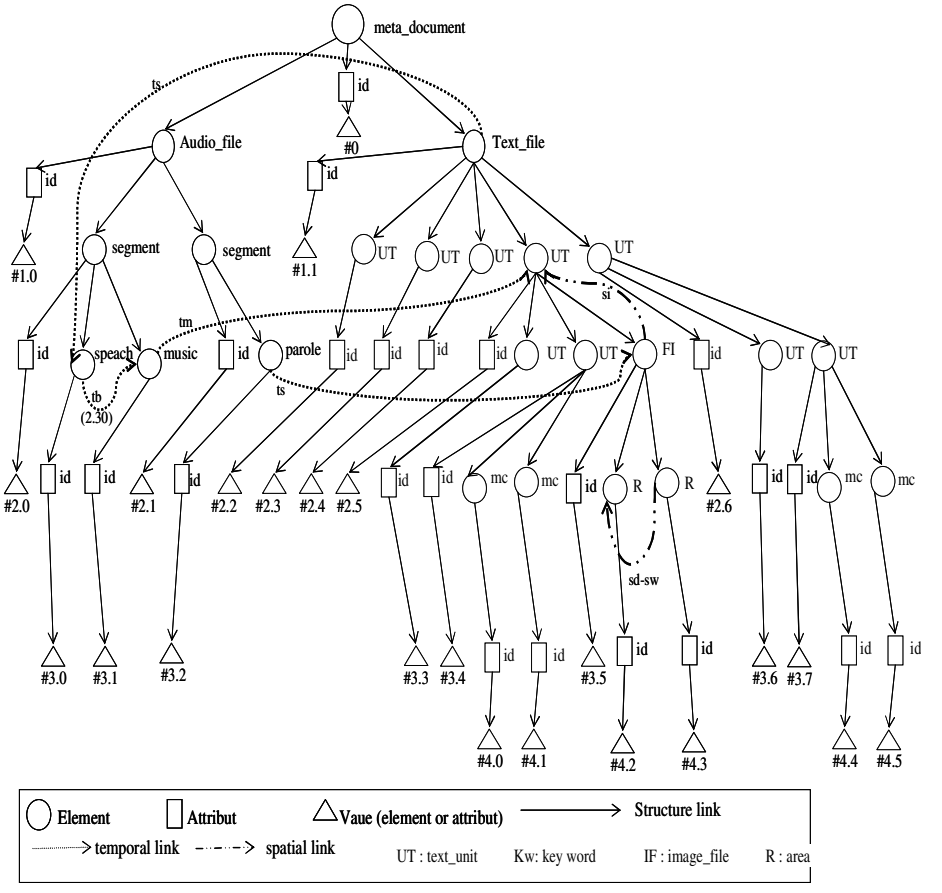


Fig. 1. The extended graph including spatial and temporal links

options. SMIL has more characteristics that are generally attractive to Web-based applications, but these are not specifically relevant to annotation. The annotation by personalised XML structure integrating spatio-temporal operators can be supported by several commercial players and easily operated in different platforms.

The annotations themselves may be either discrete or continuous, and each type may be used to annotate any other types. That is, continuous media can be associated with either discrete or continuous other media. Following this, large collections of XML marked up data have become available and the requirement for a query language to extract information from these documents is clear.

3 Multimedia Document Querying

The problem of searching XML documents can be seen as a special case of querying semi-structured data: data which have some structure but which are not stored in a fixed format like a relational database. The work to define an XML query language

has been driven largely from work in this area. A proposal has been made under the banner of the World Wide Web consortium for an XML query language called XQuery [5]. It offers better performances than older languages like XQL, XML-QL and XML-GL. It supports our requirements while exhibiting many features from previously developed languages such as the syntax for navigation in hierarchical documents from XPath and XQL, the notion of binding variables from XML-QL and the notion of functional language from OQL. XQuery is designed with the goal of being expressive, of exploiting the XML full adaptability and combining information from diverse data sources.

Since we have proposed a graphical representation of meta-documents, we have decided to apply XQuery to it. The data model supports our model by supporting the representation of collection of documents and complex value [8].

Our purpose here is to show how we can retrieve annotations using XQuery features. The queries are addressed to the meta-document. An important point to notice is that, even if the user does not know the implicit structure of the initial document. He is helped by the meta-document, of which the representation can be considered as a data guide.

Query: extract the text unit of document calling "schema description" which begins by a speech segment commented by JOHN?

```
XQuery version "1.0";
<toc>
{
for $f in doc("meta.xml")/meta_document/text_file
for $a in $f//text_unit
let $t:=$a//ST_LINKS/@link
let $b:=xs:string($a)
let $c:=xs:string("schema description ")

for $att in $a//@*
let $at:=xs:string($att)

for $spe in doc("meta.xml")/meta_document//speech
for $lt in
doc("meta.xml")/meta_document/ST_LINKS/TEMPORAL_LINK

where (contains($b, $c ) or contains($at, $c ))
            and ($spe/@speaker="JOHN")
            and ($lt/@id2 = $a/@id)
            and ($lt/@link ="ts")
            and ($lt/@id1= $spe/@id)
            and (xs:string($lt/@id2)= xs:string($a/@id))
return


<resultat>
          {
          for $i in $a/text_unit
          where ($i//key_word[@kw="schema description"])
          return
            <Title> { $i/text() }</Title>
          }
</resultat>
}
</toc>
```

4 Proposition for Adaptable Multimedia Documents by the Extended OOHDM

Once the documents are queried, we want to retrieve the adaptable multimedia documents responding to the user needs. The adaptive hypermedia is our new direction answering the traditional limitation of the static hypermedia applications. In fact, the Adaptive Hypermedia Systems (SHA) provides a support of navigation and adaptive content; which offers to every user personalized links and contents according to his interests, objectives and knowledge.

To create some dynamic pages according to the user's needs, we propose first to use elicited and instantiated metadata. The creation of dynamic views will be used to reconstitute pages containing data and metadata answering to the user's query (cf. query in section3). The link between the different views will be achieved by XPath, XLink and XPointer. Then, we propose to enrich one of the hypermedia design methodology (OOHDM [12]) by our suggested metadata approach. The extension of the methodology is proposed in order to improve the content, the structure and the access to web sites.

Extensions proposed for the OOHDM methodology are detailed in [2]. This extension consists in introducing metadata in:

- The conceptual model of the application to improve the document content on the one hand and to facilitate their querying on the other hand,
- The navigational design permitting to define the navigation between the rebuilt documents based on metadata,
- The design of the new document contents. In fact, metadata like the temporal and spatial links and others are being data representing one part of the documents.

Responding dynamically a query consists in constructing a textual unit aggregation, segments and sequences corresponding to homogeneity constraints. We propose to create different document views by integrating metadata and taking into account the user's preferences, the information he needs and his interests. Every document can be represented by one, two or several views. It is then necessary in the last case to create dynamic links between the created views to permit the navigation from a view to the other via Xpath, Xlink and Xpointer.

For the adaptable document generation, we are based on two query types:

- Displaying, for every document, the data and metadata that the user asks for,
- Displaying, for every document, the documentary unit answering to the user's query and their metadata.

4.1 Adaptable Documents by Documentary Units and Metadata

For this query type, every document contains only the applicable documentary units for the query as well as their respective metadata, if the user asks to display them. The documentary units can be represented by one, two or more views according to the user's needs. The content of each one is displayed with as a SMIL presentation as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 2.0//EN"
"http://www.w3.org/2001/SMIL20/SMIL20.dtd">

<smil xmlns=http://www.w3.org/2001/SMIL20/Language
       xmlns:xlink="http://www.w3.org/1999/xlink   >
  <body>
  <par>
    <seq>
       <par>
            <TEXT id="1" src=”ut1.txt” />
            <xlink:A   xml:link="simple"
              href="http://www.irit.fr/doc.sect#2.1"> link to
              sect#2.1 of the initial document </xlink.A>
            <xlink:simple
              href="http://www.irit.fr/prg.affich_ut?n°=1#xpoint
              er (smil/text[@id=”2”]) "/>
       </par>
       <par>
            <TEXT id="2" src=”ut2.txt” />
             <xlink:A   xml:link="simple"
              href="http://www.irit.fr/doc.sect#2.3"> link to
              sect#2.3 of the initial document </A>
             <xlink:simple
              href="http://www.irit.fr/prg.affich_ut?n°=1#xpoint
              er (smil/text[@id=”1”) "/>
       </par>
    </seq>
   <xlink:A   xml:link="simple"
  href="http://www.irit.fr/prg.mut"> Metadata page </A>
  </par>
  </body>    </smil>
```

In this example, we supposed that it subsists only two documentary units answering to the query. So, the metadata of these textual units are the title and the key words of each one.

The content of the second view containing metadata (like asked in the query of the section 3) is displayed then with XML as follows:

```
  <?xml version="1.0"?>
  <DOC>
      <METADATA>
       <MD-UT id_ud="sect #2.1" doc=" http://www.irit.fr/doc" >

          <TitleUT>dtd1</TitleUT>
          <key_word kw_1="word1" kw_2="word2" />
       </MD-UT>
       <MD-UT id_ud="sect #2.3" doc=" http://www.irit.fr/doc" >

          <TitleUT>dtd1</TitleUT>
          <key_word kw_1="word4" kw_2="word5" />
       </MD-UT>
      </METADATA>
      <A xml:link="simple"
  href="http://www.irit.fr/prg.affich_ut"> TU Page </A>
  </DOC>
```

4.2 Adaptable Multimedia Document by Metadata

For this query type, every document contains the document data and its metadata, if the user asks to display them. All data and metadata can be represented on one, two or more views according to the user's needs. The content of each one is displayed with XML as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 2.0//EN"
"http://www.w3.org/2001/SMIL20/SMIL20.dtd">
<smil xmlns=http://www.w3.org/2001/SMIL20/Language
       xmlns:xlink="http://www.w3.org/1999/xlink
       xmlns:metad="ftp:the_structure:metadata"
       >
  <body>
  <par>
     <seq>
       <text id="1" src=”ut1.txt” />
       <text id="2" src=”ut2.txt” />
       <text id="3" src=”ut3.txt” />
        <text id="4" src=”ut4.txt” />
     <seq>
     <metad:METADATA>
        <metad:TitleUT>dtd1</metad:TitleUT>
        <metad:key_word  kw_1="word1"  kw_2="word2"
  kw_3="word3"  kw_4="word4" />
    </metad:METADATA>
   <xlink:A xml:link="simple"
  href="http://www.irit.fr/prg.affich"> Previous Page </A>
  </smil>
```

5 Implementation

To validate our proposition, we have implanted a prototype called PRETI to validate our propositions concerning the improvement of integration results via the metadata integration.

The chosen application concerns tourism and culture in midi-Pyrenees (south France). The PRETI objectives are to answer to the user's needs like: looking for interesting sites, organising, considering constraints, to satisfying a maximum of identified goals, organising a journey on several days, etc.

In a first time, functionalities are:

- Flexible interrogation permitting to express the user's preferences, and
- Querying by the example allowing the user to express his query according to his needs.

Our database contains 700 resting-places; each one is described with 25 attributes that can be binary, discreet or numeric.

In a second time, functionalities of XML querying have been integrated on the platform to query resting-places and their metadata. It is these functionalities that we are going to present in this section.

At present, only textual documents have been integrated on the platform. The realization on the other type of media is to come. The step "extraction" is achieved while using two main phases:

- The phase treatment, that consists in transforming documents in another format readable by the used sensor,
- The phase of analysis, that consists in indexing meadow-treated texts while using some very specific extraction rules. The extracted metadata are used therefore to enrich the database and to ameliorate the relevant document results.

The two step "querying of created documentary basis and restitution of results" are achieved from the result of the previous step. After rewriting metadata in XML, we have used the request language of XQuery to be able to query associated data and metadata.

We have chosen to structure data of resting places and their metadata in the same document, in a homogeneous way.

The user can choose different elements that he wants to see displayed and at the same time to formulate his query while adding conditions on certain elements of resting places, including the extracted metadata. The number of views created for every answer depends on the selection made by user. The content of every constructed view or document is an XHTML document.

We want to precise also that our approach using metadata in queries presents improvement of the relevant document results about 26.5% like presented in the following table.

Table 1. Metadata contribution value in querying

Queries	Responses
Without metadata	24,50%
With metadata	51%
Improvement	26,50%

6 Conclusion

We proposed an approach of annotating multimedia document via a set of metadata specific to each type of information (text, audio, image, video), thus contributing to the extension of existing metadata for multimedia document description. These elicited metadata are structured, modeled and integrated in XML structure. For enhancing our multimedia document annotating proposal, we integrate spatial and temporal operators into annotation process. This idea enables different annotation structure for one document, witch is important to well representing the user's needs and interests. For querying XML meta-document, we choose the XQuery language.

Once the documents are extracted, they can be viewed and adapted to the user needs and performances. The generated documents can be tagged by SMIL. The document content can be enriched by metadata and can be contained only the element

interesting the users. These elements are text, audio, image or video. To enrich these multimedia documents, we have respect the OOHDM steps with its extension by metadata.

Our future works are focused to improve the querying process. We wish assist users at tow moment: on editing the query and on presenting the result. For the first assistance we develop software enabling the query edition by graphical icon. It allows also rewrite, compile and execute queries. For the second assistance, we can create a mediator between the graphical query edition software and a commercial SMIL editor.

References

[1] Allen, J.F.: Time and Time Again: The Many Ways to Represent Time. International Journal of Intelligent Systems 6(4), 355–391 (1991)

[2] Amous, I., Chrisment, C., Sèdes, F.: Reengineering the Web sites by using metadata and a methodological approach. In: 3rd International Conference on Information Integration and Web-based Applications & Services – iiWAS 2001, Linz, Australie, Septembre 10-12, pp. 127–138 (2001)

[3] Amous, I., Jedidi, A., Sèdes, F.: A contribution to multimedia modelling and querying. In: Multimedia Tools and Applications (MTAP), vol. 25, pp. 391–404. Kluwer academic publishers, Dordrecht (2005)

[4] Bird, S., Buneman, P., Tan, W.C.: Towards a query language for annotation graphs. In: Proc. of LREC, Athens, Greece (2000)

[5] Chamberlin, D.: XQuery: A Query Language for XML. In: Proc. SIGMOD Conference, San Diego, United States, June 9-12, p. 682 (2003)

[6] Dorado, A., Izquierdo, E.: Semi-Automatic Image Annotation Using Frequent Keyword Mining. In: Proc. 7th International Conference on Information Visualization, London, England, July 16-18, p. 537 (2003)

[7] Djeraba, C.: Multimedia mining, a highway to intelligent multimedia documents. Kluwer Academic Publishers, Dordrecht (2003)

[8] Fernandez, M., Marsh, J., Nagy, M.: XQuery1.0 & XPath2.0 Data Model, W3C, Retrieved (November 2002), http://www.w3c.org/TR/query-datamodel/

[9] Jaffré, G., Joly, P.: Costume: a New Feature for Automatic Video Content Indexing. In: Dans: RIAO 2004 Coupling approaches, coupling media and coupling languages for information retrieval, Avignon, April 26-28, pp. 314–325 (2004)

[10] Lementini, E.C., Felice, P., Oosterom, D., Van, P.: A Small Set of Formal Topological Relationships Suitable for End-User Interaction. In: Abel, D.J., Ooi, B.-C. (eds.) SSD 1993. LNCS, vol. 692, pp. 277–295. Springer, Heidelberg (1993)

[11] Millard, D.E., et al.: Hyperdoc: An Adaptive Narrative System for Dynamic Multimedia Presentations. Technical Report, ECSTR-IAM02-006 (2003), http://eprints.ecs.soton.ac.uk/7279/

[12] Schwabe, D., Emeraldo, L., Rossi, G., Lyardet, F.: Engineering web aplications for reuse. IEEE Multimedia 8(2) (2001)

Shifting Predicates to Inner Sub-expressions for XQuery Optimization

Sven Groppe[1], Jinghua Groppe[2], Stefan Böttcher[3], and Marc-André Vollstedt[3]

[1] University of Lübeck, Institute of Information Systems, Ratzeburger Allee 160, D-23538 Lübeck, Germany
groppe@ifis.uni-luebeck.de
[2] Kahlhorststrasse 36a, D-23562 Lübeck, Germany
[3] University of Paderborn, Fürstenallee 11, D-33102 Paderborn, Germany
stb@upb.de, marc-andre.vollstedt@gmx.de

Abstract. Besides XPath, XQuery is an XML query language developed by W3C. Within this paper, which is an extended version of [6], we present an approach for the optimization of the execution time of nested XQuery queries based on type information of the input XML document given in XML Schema. The optimization approach deletes unnecessary sub-expressions and shifts predicates to inner sub-expressions wherever possible in order to avoid the generation of unnecessary intermediate data. A performance analysis of our prototype demonstrates the optimization of the execution time of various current XQuery evaluators when using the proposed optimization approach.

1 Introduction

Many XML databases and XML-enabled databases support XQuery [15] as a query language for querying XML documents. Furthermore, XQuery evaluators have been implemented as stand alone engines to access XML documents stored in files. This paper presents a method, the purpose of which is to optimize the execution times of XQuery queries.

We describe how to eliminate sub-expressions, the result of which is not used for the computation of the final result of an XQuery expression. Furthermore, we show how to shift predicates to inner sub-expressions so that only those XML nodes are retrieved that are necessary for the computation of the final result. XQuery queries containing sub-expressions that specify unnecessary intermediate results can be (a) a user-defined query when users do not optimize manually or (b) an automatically generated query for example (1) in the scenario of queries on views that can be represented as nested queries, or (2) in the scenario of access control, where the access right of an user is executed as inner query and the user query as outer query.

We show by experimental results that current implementations do not use our proposed optimization rules and that these implementations improve their performance when using our proposed optimization rules.

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 67–79, 2009.

2 Optimization Approach

The outline of this section is as follows: Section 2.1 presents an example of a non-optimized query and its optimized query. Section 2.2 describes how to represent the XQuery query as a graph that represents the information about the possible output of the XQuery query. Section 2.3 describes how to search in this graph representation of the XQuery query according to XPath expressions that are embedded in the XQuery query. The result of the search describes those sub-expressions of the XQuery query that are used for the computation of the final result of the XQuery query. In Section 2.4, we describe how to optimize the XQuery query by eliminating those sub-expressions that are not used for the computation of the final result. Section 2.5 presents how to shift predicates to inner sub-expressions so that predicates can be directly applied to the input XML document (instead of computing intermediate results). Section 2.6 presents the optimization algorithm in pseudo code.

2.1 Example

For an example, see Figure 1 to Figure 3. The queries of the example can be executed with input XML documents of the XMark benchmark [13]. Figure 1 contains the non-optimized query of the example, where the inner query from line 1 to line 6 represents the view (or the access right for the user respectively) and the outer query in line 7 represents the query on the view in the scenario of queries on views (or the user query respectively in the scenario of access control). Figure 2 contains the example query after applying the approach presented in Section 2.4 that eliminates those sub-expressions that are not necessary for the computation of the final result. Figure 3 contains the overall optimized query after applying the approach presented in Section 2.5 that shifts predicates to inner sub-expressions.

```
1  let $view := <root> <result> <preferred>
2    {/child::site/child::people/child::person/child::profile}</preferred>
3   <standard>{/site/regions/australia/item}</standard>
4   <challenge>{/child::site/child::open_auctions/child::open_auction}</challenge>
5   <na>{for $p in /site/people/person where $p/descendant-or-self::person
6        return $p}</na> </result> </root>
7  return $view/child::result/child::na/child::person[attribute::id="person007"]
```

Fig. 1. Non-optimized query, which is also query 2 in the experiments

```
let $view := <root> <result> <na> {for $p in /site/people/person
 where $p/descendant-or-self::person return $p} </na> </result> </root>
return $view/child::result/child::na/child::person[attribute::id="person007"]
```

Fig. 2. Reduced XQuery query

```
let $view := <root> {let $i1 := <result> {let $i2 := <na>
 {for $p in /site/people/person
  where $p/descendant-or-self::person and $p[attribute::id="person007"] return $p}
  </na> where $i2/child::person[attribute::id="person007"] return $i2}</result>
  where $i1/child::na/child::person[attribute::id="person007"] return $i1}
 </root>
return $view/child::result/child::na/child::person[attribute::id="person007"]
```

Fig. 3. Optimized query, where predicates are shifted to inner sub-expressions

2.2 XQuery Graph

The *XQuery graph* is generated from the XQuery query and represents the possible output of the XQuery query.

For the optimization rules, we consider that subset of XQuery, where the XQuery expression must conform to following rule `Start` in EBNF notation.

```
Start ::= (FunctionDecl)* FLRExpr.
FunctionDecl::="declare" "function" QName "(" ("$" QName ("," "$" QName)*)? ")"
              "{"ExprSingle"}".
FLRExpr ::= (ForClause | LetClause)+ "return" ExprSingle.
ForClause ::= "for" "$" VarName "in" ExprSingle.
LetClause ::= "let" "$" VarName ":=" ExprSingle.
ExprSingle ::= FLRExpr|IfExpr|PathExpr.
IfExpr ::= "if" "(" ExprSingle ")" "then" ExprSingle "else" ExprSingle.
PathExpr ::= ("/" RelativePathExpr?) | ("//"RelativePathExpr)  | RelativePathExpr.
RelativePathExpr ::= (Step | PrimaryExpr)(("/"|"//") (Step | PrimaryExpr))*.
Step ::= (("child" | "descendant" | "attribute" | "self" | "descendant-or-self" |
          "following-sibling" | "following" | "parent" |"ancestor" |
          "preceding-sibling" | "preceding" | "ancestor-or-self") "::")?
        (QName | "node()" | "*").
PrimaryExpr ::= "$" QName | Constructor | FunctionCall.
Constructor ::= ("element" | "attribute") QName "{" ExprSingle "}".
FunctionCall ::= QName "(" (ExprSingle ("," ExprSingle)*)? ")".
```

This subset of XQuery contains nested `for-let-return` clauses, `if` expressions, element and attribute constructors, declarations of functions and function calls.

For example, the XQuery graph of the query of Figure 1 is presented in Figure 4. Note that in the example of Figure 4, we use the schema information of the XMark benchmark [13].

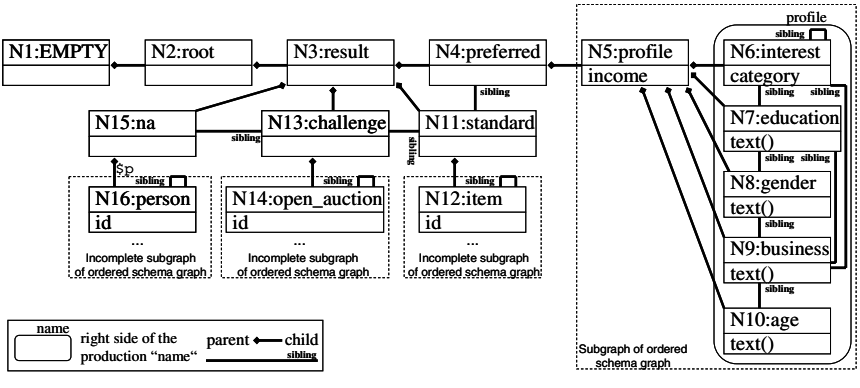


Fig. 4. XQuery graph of Figure 1

Each node of the XQuery graph represents an element node that can be generated by the XQuery query or a dummy node (called `:EMPTY` node). The `:EMPTY` node represents a whole sub-expression and is a shortcut for the sub-expression. There are three kinds of edges: *parent-child edges* represent a parent child relationship between element nodes and/or `:EMPTY` nodes. A *sibling edge* represents a directed sibling relationship between element nodes and/or `:EMPTY` nodes. Finally, an *expression*

edge represents a relationship to a whole sub-expression of the XQuery query between element nodes and :EMPTY nodes.

Whereas the detailed algorithm for the generation of the XQuery graph is presented in the Appendix (see Section 7), we present the general rules for generating the XQuery graph from an XQuery expression: We generate an own XQuery graph for each variable assignment $view of the XQuery expression. Every expression within the variable assignment of $view, which generates output, gets its own new node N representing the output. This new node N becomes the parent node of every node in the XQuery graph representing output, which could be generated as child node of node N by the XQuery evaluator. Furthermore, we draw a sibling edge from the new node N to every node N2 in the XQuery graph representing output, which could be generated as sibling node of node N by the XQuery evaluator. Usages of variables are replaced with the content of the corresponding variable assignment.

If the next generated output of the XQuery evaluator is a sub-tree of the input XML document specified by an XPath expression XP, we first compute the *ordered schema graph* of the schema definition of the input XML document. Subgraphs of the ordered schema graph are then integrated into the XQuery graph in order to represent the schema of that part of the input XML document that is output of the XQuery query.

Ordered Schema Graph. Whereas the detailed algorithm for the generation of the ordered schema graph is omitted here due to space limitations, we present the general rules for generating the *ordered schema graph* from an XML Schema definition:

- We create a start node of type :EMPTY, the child of which is a node representing that element, which is the root node of the XML document.
- We add all (required, implied and fixed) attributes of any type of the corresponding XML element E to the nodes representing E.
- We transform an element declaration according to the following rules:
  - Nodes of the ordered schema graph representing elements are the parents of the nodes representing their inner elements specified in the element declaration.
  - Whenever an element E1 can be a following sibling node of another element E2, we insert a sibling edge from E2 to E1. This is the case for repetitions of elements and for sequences of elements.
  - Whenever the XML Schema defines an element to be a complexType defined by a choice or a sequence, then we create an extra :EMPTY node for easy access to the whole choice expression or the whole sequence expression, respectively.

We come back to the generation of the XQuery graph. If the next generated output of the XQuery evaluator is a sub-tree of the input XML document specified by an XPath expression XP, we search within this ordered schema graph by the modified XPath evaluator as stated in Section 2.3, and we retrieve a node set SN of nodes of the ordered schema graph. We first copy the nodes of SN and copy all descendant nodes and all sibling nodes, which can be reached from the nodes of SN by parent-child relationships and sibling relationships. We copy also all parent-child relationships and sibling relationships for the copied nodes. Finally, we set the current node as parent

node of the copies of SN. We label the association with the XPath expression of the XQuery expression.

2.3 Output Search

Whenever the content of a variable $view is queried by an XPath expression XP by $view/XP in the XQuery query, we search in that part of the XQuery graph that represents the output of the variable $view. The search can be executed by a modified XPath evaluator, which executes the XPath query XP on the XQuery graph instead of an XML document.

Within the XQuery graph, there is all necessary information in order to execute the modified XPath evaluator, as the parent-child-axis and the next-sibling-axis are available in the XQuery graph. The evaluator passes empty nodes without consuming an element of the XPath expression XP. In comparison to an XML document, the XQuery graph can contain loops. Therefore, the modified XPath evaluator must consider loops when it revisits a node but did not process the next location step within XP. For this purpose and for the succeeding optimization steps, the modified XPath evaluator marks all the nodes that contribute to a successful evaluation of XP and stores them into *detected paths* of the XQuery graph.

We execute the modified XPath evaluator on the XQuery graph of the variable assignment of $view with the input XPath query XP. In the case of the XQuery expression in Figure 1, the XPath query XP is `/child::result/child::na/child::person[attribute::id="person007"]` for $view.

2.4 Reducing the XQuery Query

This section outlines how we can reduce the XQuery query according to the marked nodes of the XQuery graph of the output search so that only those sub-expressions remain that are necessary for the computation of the final result.

Within the optimization step, we do not delete all the sub-expressions within the XQuery expression from which the marked nodes in the XQuery graph are generated and which do not assign variables, which are used in sub-expressions of marked nodes. The for-statement defines how often and in which order the result-statement is executed. Therefore, we do not delete for-statements, except if their return-statement is reduced to an empty statement.

We delete all other unmarked sub-expressions and finally a partially optimized query remains. The following Section 2.5 describes how to shift predicates to inner sub-expressions of the XQuery query.

2.5 Shifting Predicates to Inner Sub Expressions

Whereas we have described in Section 2.4 how we can eliminate those XQuery *sub-expressions*, which are not needed for the final result, we describe in this section how we can insert predicates into the XQuery query so that only those *XML nodes* are retrieved, which are needed to compute the final result.

For this purpose, we restrict the retrieved node set of each variable and of each retrieved node set described by an XPath expression, which is not directly assigned to a

variable. This is the case for all XQuery expressions **for** $var **in** XQ1 XP XQ2 and **let** $var := XQ1 XP XQ2, where XP is an XPath expression and XQ1 or XQ2 are expressions, which can generate output. XQ1 or XQ2 respectively generate output if XQ1 or XQ2 respectively contain XML node constructors.

Then we consider each variable $var. We determine those tuples t of detected paths and their branched paths, the second entry of which represents the content of the variable $var. We determine all those remaining XPath expressions XP1, …, XPn, which are the first entry of the tuples of t. If XP1, …, XPn contain only forward axes, i.e. child, descendant-or-self, descendant, self, attribute, following or following-sibling, then we can restrict the node set of the variable $var by inserting the **where**-clause **where** XP1 **or** … **or** XPn. If an XPath expression XPi in XP1, …, XPn contains a reverse axis, i.e. parent, ancestor, ancestor-or-self, preceding or preceding-sibling, then XPi describes a node set, which can contain XML nodes, which are not stored in $var and which have been computed in the XQuery expression before $var is accessed. In some cases, we can avoid that XP1, …, XPn contain a reverse axis by applying the approach presented in [9] to all XPath expressions, which access variables, before we mark the nodes in the XQuery graph. The approach in [9] eliminates all reverse axes in an XPath expression so that the computed equivalent XPath expression contains only forward axes. We refer to [9] for details.

We analogously restrict the retrieved node sets of XPath expressions XP, which are not directly assigned to a variable. We determine all those XPath expressions XP1, …, XPn, which are the remaining XPath expressions to be searched for in the detected paths of the output search in the XQuery graph. If XP1, …, XPn only contain forward axes, then we modify those XPath expressions XP to XP[XP1 **or** … **or** XPn]. Again, we can avoid that one of XP1, …, XPn contains a reverse axis for some cases by applying the approach of [9] first, before we mark the nodes in the XQuery graph. We refer to [9] for details.

2.6 Optimization Algorithm in Pseudo Code

```
Algorithm OptimizeQuery
Input: XQuery query Q conforming to rule Start in Section 2.2
Output:Optimized XQuery query Q'

(1) Generate abstract syntax tree T of Q
(2) Compute XQuery graph XG with marked nodes of T
(3) Mark all nodes in T, which correspond to marked nodes in XG
(4) while(all children of a symbol ExprSingle of a LetClause expression are unmarked) do
(5)  delete the whole LetClause expression
(6) For all nodes n in T do
(7)  If(n and its child nodes are unmarked and (n is a symbol ExprSingle and
        not(n is a parameter of a function call or n is a condition of an if-statement))) then
(8)   delete n (and its children)
(9)  If(n is marked and (n is an XPath expression or n is a ForClause or a LetClause)) then
(10)  Let XP1, …, XPi be the remaining XPath expressions of the detected paths containing n
(11)  If(XP1, …, XPi contain only forward axes) then
(12)   If(n is a ForClause or a LetClause of a variable $var) then
(13)    If(n has a where-clause where B) then
(14)     Replace B with B or $var[Xp1 or … or XPi]
(15)    else insert at n a where-clause "where $var[Xp1 or … or XPi]"
(16)   If(n is an XPath expression XP) then replace XP with XP[Xp1 or … or XPi]
(17) Compute Q' from the remaining nodes of T
```

Fig. 5. Optimization algorithm in pseudo code

Figure 5 contains the optimization algorithm in pseudo code. The input of the optimization approach is the non-optimized query `Q` that must conform to the rule `Start` given in Section 2.2. Lines (1) to (8) contain the pseudo code of the approach presented in Section 2.4, and Lines (9) to (16) the pseudo code of the approach presented in Section 2.5. Line (17) contains the command to compute the optimized query `Q'`.

3 Performance Analysis

3.1 Experimental Environment

The test system for the experiments uses an AMD Athlon 1800+ processor, 768 Megabytes DDR RAM, runs Windows 2000 and Java version 1.4.2. We have chosen Saxon version 8.4 [7] and Qexo version 1.7.90 [12] as XQuery evaluators. We use the input XML documents up to 5 Megabytes in 0.5 Megabytes steps generated by the data generator of the XMark Benchmark [13]. Furthermore, we use following queries for the experiments. Query 1 is presented in Figure 6, Figure 1 contains query 2, query 3 is presented in Figure 7, Figure 8 contains query 4 and query 5 is presented in Figure 9.

```
for $view := <root>
{for $i in /child::site/child::people/child::person/child::profile/child::interest
  let $p := for $t in /child::site/child::people/child::person
   where $t/child::profile/child::interest
   return <personne><statistiques><sexe>
         {$t/child::gender}</sexe><age>{$t/child::age}</age>
         <education>{$t/child::education}</education>
         <revenu>{$t/child::income}</revenu></statistiques>
         <coordonnees><nom>{$t/child::name}</nom><rue>{$t/child::street}</rue>
         <ville>{$t/child::city}</ville><pays>{$t/child::country}</pays>
         <reseau><courrier>{$t/child::email}</courrier>
         <pagePerso>{$t/child::homepage}</pagePerso></reseau></coordonnees>
         <cartePaiement>{$t/child::creditcard}</cartePaiement></personne>
  return <categorie>{<id>{$i}</id>}{$p}</categorie>}
</root> return $view/categorie/personne/cartePaiement/creditcard
```

Fig. 6. Query 1

```
let $view := <root>{for $i in
 /child::site/child::regions/child::australia/child::item
 return <item>{$i/descendant-or-self::item}</item>}
<root> return $view/child::item/child::item[attribute::id="item11"]
```

Fig. 7. Query 3

```
let $view := <root>{for $b in
 /child::site/child::open_auctions/child::open_auction
 where $b/child::bidder/child::personref[attribute::person="person0"]
 return <history>{$b/child::itemref}</history>}
</root> return $view/child::history/child::itemref[attribute::item="item007"]
```

Fig. 8. Query 4

```
let $view := <root>{for $b in /site/regions/asia return $b/child::item}
</root> return $view/child::item[attribute::id="item2"]
```

Fig. 9. Query 5

3.2 Analysis of Experimental Results

The sizes in Kilobytes of the intermediate results stored in the variable `$view` of the non-optimized queries and their optimized queries are presented in Figure 10. Furthermore, Figure 10 contains the size of `$view` of the optimized query divided by the size of `$view` of the non-optimized query in percentage, which we call the *selectivity of the optimization*. The speed up factor is defined to be the execution time of the optimized query divided by the execution time of the non-optimized query. Figure 11 presents the speed up factors of the queries 1 to 5 and Figure 12 presents a zoom of Figure 11 for the queries 2 to 5.

query \ file size	1 MB	2 MB	3 MB	4 MB	5 MB
1, non-optimized	7560	33738	77104	122752	215414
1, optimized	575	2491	6011	9241	16317
1, select. in %	7.61%	7.38%	7.80%	7.53%	7.57%
2, non-optimized	414	803	1191	1624	2069
2, optimized	1	1	1	1	1
2, select. in %	0.24%	0.12%	0.08%	0.06%	0.05%
3, non-optimized	41	93	144	200	239
3, optimized	1	7	2	2	4
3, select. in %	2.44%	7.53%	1.39%	1%	1.67%
4, non-optimized	1	1	1	1	1
4, optimized	1	1	1	1	1
4, select. in %	100%	100%	100%	100%	100%
5, non-optimized	31	88	137	163	218
5, optimized	2	1	5	2	2
5, select. in %	6.45%	1.14%	3.65%	1.23%	0.92%

Fig. 10. The sizes in Kilobytes of the intermediate results stored in the variable `$view` of the non-optimized queries and their optimized queries and the selectivity of the optimization

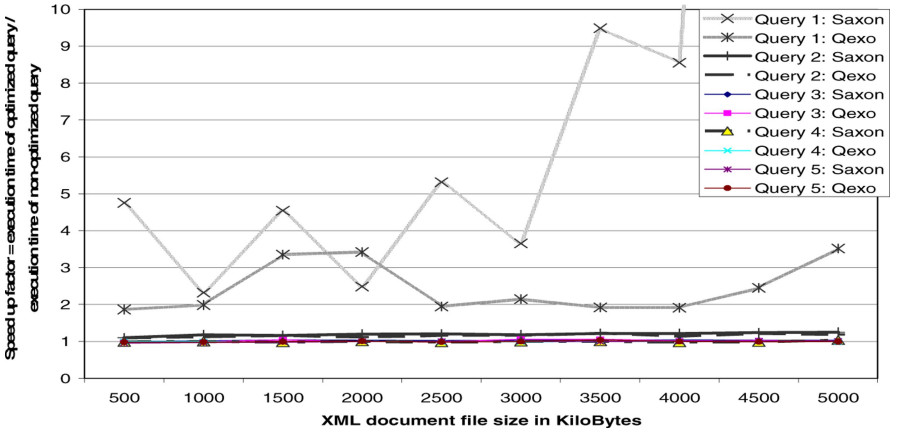


Fig. 11. Speed up factors of the queries 1 to 5

Considering also Figure 10, we notice that as smaller the selectivity of the optimization is and as larger the size of the intermediate results of `$view` are, as bigger are the speed up factors. Furthermore, although the selectivity of the optimization is 100% and thus we cannot optimize by our approach, the speed up factor is close to 1 and we are not much slower compared to not using our optimization approach.

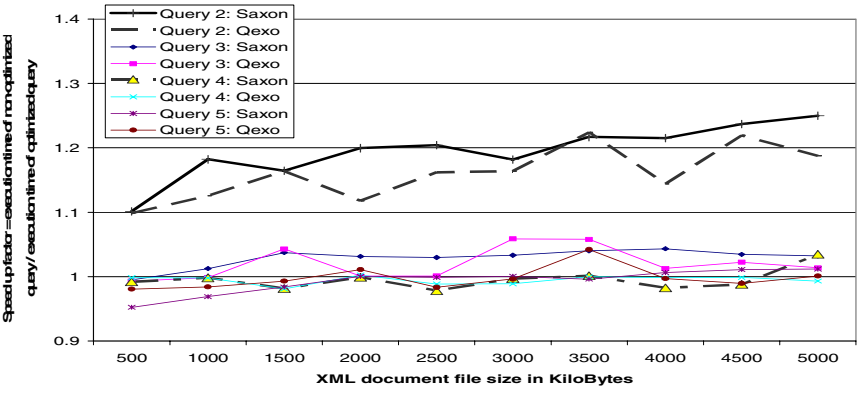


Fig. 12. Zoom of Figure 11, speed up factors of the queries 2 to 5

4 Related Work

The contributions [1], [2] and [11] introduce an algebra for XQuery. Additionally, they list transformation and optimization rules based on the introduced algebra, but they do not contain the optimization approach presented in this paper.

[8] projects XML documents to a sufficient XML fragment before processing XQuery queries. It contains a static path analysis of XQuery queries, which computes a set of projection paths formulated in XPath. Our approach optimizes the XQuery expression itself and does not project XML documents.

Papakonstantinou et al. [10] studies the inference of DTDs for views of XML data, but uses the language loto-ql for the definition of XML views, which is less powerful than XQuery. Furthermore, our approach optimizes all XQuery queries and cannot only be used in the scenario of XQuery queries on XQuery views.

Our work was inspired by the contribution in [5], which deals with XPath query reformulation according to an XSLT view and its optimization. In comparison, in this paper, we describe general optimization rules, which are especially applicable for query reformulation on an XQuery view. We extend our work in [4] by an approach to shift predicates to inner sub-expressions.

In comparison to all other approaches, we focus on the optimization of XQuery queries based on the schema of the input XML document in order to eliminate unnecessary query code and to shift predicates to inner sub-expression so that the optimized query avoids the computation of non-used intermediate results.

5 Summary and Conclusions

In this paper, we have examined optimization rules, the goals of which are to eliminate those sub-expressions in an XQuery query that are not necessary for the computation of the final result and to shift predicates to inner sub-expressions. These optimization rules are very useful for example in the scenario of XQuery queries on XQuery views. First, we introduce the XQuery graph of an XQuery expression `Q` representing the output of `Q`. Second, we search in variable assignments of XQuery queries for all sub-expressions that are used for the computation of the final result. Afterwards, we eliminate all other sub-expressions. Furthermore, we shift predicates to inner sub-expressions so that predicates are executed as early as possible.

We have shown by experimental results that the evaluation of the optimized queries saves processing costs depending on the amount of saved unnecessary intermediate results. In the case that our approach cannot optimize the XQuery query, we are not much slower compared to not using our optimization approach.

We are of the opinion that it is promising to investigate further optimization rules for XQuery queries.

References

1. Fisher, D., Lam, F., Wong, R.K.: Algebraic transformation and optimization for xQuery. In: Yu, J.X., Lin, X., Lu, H., Zhang, Y. (eds.) APWeb 2004. LNCS, vol. 3007, pp. 201–210. Springer, Heidelberg (2004)
2. Grinev, M., Kuznetsov, S.: Towards an exhaustive set of rewriting rules for xQuery optimization: BizQuery experience. In: Manolopoulos, Y., Návrat, P. (eds.) ADBIS 2002. LNCS, vol. 2435, pp. 340–345. Springer, Heidelberg (2002)
3. Groppe, S.: XML Query Reformulation for XPath, XSLT and XQuery. Sierke-Verlag, Göttingen (2005)
4. Groppe, S., Böttcher, S.: Schema-based query optimization for XQuery queries. In: Eder, J., Haav, H.-M., Kalja, A., Penjam, J. (eds.) ADBIS 2005. LNCS, vol. 3631. Springer, Heidelberg (2005)
5. Groppe, S., Böttcher, S., Birkenheuer, G., Höing, A.: Reformulating XPath Queries and XSLT Queries on XSLT Views. Data and Knowledge Engineering 57, 64–110 (2006)
6. Groppe, S., Groppe, J., Böttcher, S., Vollstedt, M.-A.: Shifting Predi-cates to Inner Sub-Expressions for XQuery Optimization. In: Damiani, E., Yetongnon, K., Chbeir, R., Dipanda, A. (eds.) SITIS 2006. LNCS, vol. 4879, pp. 67–79. Springer, Heidelberg (2009)
7. Kay, M.H.: Saxon - The XSLT and XQuery Processor (April 2004), `http://saxon.sourceforge.net`
8. Marian, A., Siméon, J., Projecting, X.M.L.: Documents. In: Proceedings of the 29th VLDB Conference, Berlin, Germany (2003)
9. Olteanu, D., Meuss, H., Furche, T., Bry, F.: XPath: Looking Forward. In: XML-Based Data Management (XMLDM), EDBT Workshops, Prague, Czech Republic (2002)
10. Papakonstantinou, Y., Vianu, V.: DTD Inference for Views of XML Data. In: Proceedings of the Nineteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2000), Dallas, Texas, USA (2000)

11. Paparizos, S., Wu, Y., Lakshmanan, L.V.S., Jagadish, H.V.: Tree Logical Classes for Efficient Evaluation of XQuery. In: SIGMOD 2004, Paris, France (2004)
12. Qexo: The GNU Kawa implementation of XQuery (2003), http://www.gnu.org/software/qexo
13. Schmidt, A., Waas, F., Kersten, M., Carey, M., Manolescu, I., Busse, R.: XMark: A benchmark for XML data management. In: Bressan, S., Chaudhri, A.B., Li Lee, M., Yu, J.X., Lacroix, Z. (eds.) VLDB 2002. LNCS, vol. 2590, pp. 974–985. Springer, Heidelberg (2003)
14. W3C, XML Path Language (XPath) Version 1.0. (1999), http://www.w3.org/TR/xpath/
15. W3C, XQuery 1.0: An XML Query Language, W3C Working Draft (2003), http://www.w3.org/TR/xquery/

Appendix

Within this section, we describe the algorithm, which generates the XQuery graph from an XQuery expression XP, and which marks the nodes of the XQuery graph in variable assignments according to the approach presented in Section 2.2 and in Section 2.3. For the description of the algorithm, we use *attribute grammars*.

Definition: An attribute grammar consists of a grammar G in EBNF notation and computation rules for attributes of symbols of G added to a production rule of G.

In the following, we use the notation P { C }, where P is a production rule of G in the EBNF notation and C contains the computation rules for attributes of symbols, which occur in P. We use a slightly different variant of the Java notation in C. We refer to an attribute a of the m-th symbol n in P as n[m].a. If there is only one symbol n in P, we use n.a instead of n[1].a. If there exists an arbitrary number of symbols n in P, then i represents the concrete number of occurrences of the symbol n in P. i1 and i2 respectively represent the concrete amounts of occurrences if two symbols n1 and n2 respectively can both occur an arbitrary number of times in P.

First, we parse a given XQuery expression XP, which generates the abstract syntax tree of XP. Then, we evaluate the following attribute grammar on this abstract syntax tree, which computes the attributes of the nodes in the abstract syntax tree. Finally, the resultant XQuery graph of an XQuery expression XP is stored in the attribute nodeSet of that node in the abstract syntax tree, which represents the symbol Start.

Most node sets stored in the attribute nodeSet of symbols in the attribute grammar are propagated from underlying symbols as e.g. in the following rule.

```
Start ::= (FunctionDecl)* FLRExpr. { Start.nodeSet=FLRExpr.nodeSet; }
```

Within the declaration of a function, we call a function addFunctionNodeSet, which stores the node set of the function body associated with the function name.

```
FunctionDecl ::= "declare" "function" QName "(" ("$" QName ("," "$" QName)*)? ")"
                 "{" ExprSingle "}". {
 FunctionDecl.nodeSet = ExprSingle.nodeSet;
 addFunctionNodeSet(QName.getString(),FunctionDecl.nodeSet); }
```

A `for` clause iterates over a node set and for each iteration step, the `return` clause is evaluated. Therefore, the nodes of the node set of the `return` clause are associated to these nodes themselves by a sibling relationship.

```
FLRExpr ::= (ForClause|LetClause)+ "return" ExprSingle. {
 FLRExpr.nodeSet=ExprSingle.nodeSet;
 if(there exists at least one ForClause)
  for all n in FLRExpr.nodeSet do n.addSiblingNodes(FLRExpr.nodeSet); }

ForClause ::= "for" "$" VarName "in" ExprSingle. {
 ForClause.nodeSet = ExprSingle.nodeSet;
 StoreVarNodeSet(VarName.getString(),ExprSingle.nodeSet); }

LetClause ::= "let" "$" VarName ":=" ExprSingle. {
 LetClause.nodeSet = ExprSingle.nodeSet;
 StoreVarNodeSet(VarName.getString(),ExprSingle.nodeSet); }

ExprSingle ::= FLRExpr. { ExprSingle.nodeSet=FLRExpr.nodeSet; }

ExprSingle ::= IfExpr. { ExprSingle.nodeSet=IfExpr.nodeSet; }
```

If an XPath expression is detected, then we call the function `getNodeSetOfXPath` with the path expression as parameter. The function `getNodeSetOfXPath` evaluates the XPath expression, marks all the successfully visited nodes in the XQuery graph, and returns a clone of the result node set and all their connected nodes.

```
ExprSingle ::= PathExpr. {
 ExprSingle.nodeSet = getNodeSetOfXPath(PathExpr.getString()); }
```

The node set of an `if` expression is the union of the node set of the expression after `then` and the node set of the expression after `else`.

```
IfExpr ::=  "if" "(" ExprSingle ")" "then" ExprSingle "else" ExprSingle. {
 IfExpr.nodeSet = ExprSingle[2].nodeSet ∪ ExprSingle[3].nodeSet; }
```

If the content of a variable is retrieved, we determine the node set of the variable by a function `getVarNodeSet`. This function `getVarNodeSet` returns the node set of the variable or returns the variable name, if the variable is a parameter of a function. In this case, the variable name will be later replaced by its corresponding node set, when we retrieve the node set of a function call.

```
PrimaryExpr ::= "$" QName. {PrimaryExpr.nodeSet=getVarNodeSet(QName.getString());}

PrimaryExpr ::= Constructor. { PrimaryExpr.nodeSet = Constructor.nodeSet; }

PrimaryExpr ::= FunctionCall. { PrimaryExpr.nodeSet = FunctionCall.nodeSet; }
```

In case of element constructors or attribute constructors respectively, an element node or attribute node respectively is created in the XQuery graph.

```
Constructor ::="element" QName "{"ExprSingle"}".{e=new Element(QName.getString());
 e.setChildren_and_Attributes(ExprSingle.nodeSet);Constructor.nodeSet.add(e); }

Constructor ::= "attribute" QName "{" ExprSingle "}". {
 Constructor.nodeSet.add(new Attribute(QName.getString())); }
```

In case of a function call, the result node set of the function is determined by a function `getFunctionNodeSet`. The called function can be a built-in function, or a user-defined function.

In case of a built-in function, the function `getFunctionNodeSet` returns a cloned node set of the possible result node set of the function. In case of the `document` function of an XML file `f`, the result node set contains the root node of `f`, which is associated with the ordered schema graph of the schema of `f`.

In case of a user-defined function, the function `getFunctionNodeSet` returns a cloned node set of the result node set of the function, where the parameter variables are instantiated with the concrete node sets of the parameter variables. This includes the evaluation of XPath expressions, which contain references to parameter variables, inside the function body. If loops are detected, we stop the further computation of the node sets of the functions and we refer in the current node set to the already computed node set of the loop. A corresponding approach is presented in [5] for XSLT templates instead of XQuery expressions.

```
FunctionCall ::= QName "(" (ExprSingle ("," ExprSingle)*)? ")". {
 FunctionCall.nodeSet = getFunctionNodSet(
    QName.getString(),ExprSingle[1].nodeSet,…,ExprSingle[i].nodeSet); }
```

AIRSTD: An Approach for Indexing and Retrieving Spatio-Temporal Data

Hatem F. Halaoui

Department of Computer Science & Mathematics
Haigazian University
Rue Mexique, Kantari, Riad El Solh, Beirut 1107 2090
Lebanon
hhalaoui@haigazian.edu.lb

Abstract. Geographical (spatial) information about the real world changes rapidly with time. We can simply see examples of these changes when we look at any area. New buildings, new roads and highways, and many other new constructions are added or updated. Spatial changes can be categorized in two categories: (1) Discrete: changes of the geometries of physical entities (i.e., buildings) and (2) abstract: moving objects like airplanes, cars or even moving people. Spatio-temporal databases need to store information about spatial information and record their changes over time. The main goal our study in this paper is to find an efficient way to deal with spatio-temporal data, including the ability to store, retrieve, update, and query. We offer an approach for indexing and retrieving spatio-temporal data (AIRSTD). We concentrate on two main objectives: (1) Provide indexing structures for spatio-temporal data and (2) provide efficient algorithms to deal with these structures.

Keywords: Spatial, Temporal, Spatio-Temporal, Geographical, Spatial indexing, Temporal indexing, Spatio-Temporal indexing.

1 Introduction

Most spatial information in the real world is also temporal information. In other words, we rarely find geographical information that does not change. Many examples can be given: new buildings are built, new roads and highways are constructed, and other new constructions are added or updated. Moreover, changes of the geometries of physical (i.e. buildings) and abstract (i.e. land parcels) entities need to be recorded. In most cases people want their systems to keep the old and the current information and sometimes make estimates of the future information. Adding a temporal dimension to the spatial database increases the complexity of the managing information.

These issues lead to the many questions:

1- How can we save all the information both past and current?
2- Do we save snapshots of the information where a snapshot has full information?
3- What is the interval of validity of each snapshot?

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 80–90, 2009.

4- What happens if a change occurs during this interval? Do we discard it and hence lose it?
5- Can we take a snapshot at each change? Is this efficient?

Analyzing these questions will lead to the following main points:

What is the most efficient way to save all the information at all times with no loss, taking into consideration these three issues:

- Having uniform intervals for efficient searching.
- Changes within intervals are recorded for consistency.
- Non-changing information is not repeated at all snapshots for space efficiency.

We can summarize these considerations with a simple question:
What is the most efficient way to store, retrieve, and query spatio-temporal data?

The paper is organized as follows: Section 2 presents some related work in the area of spatial and spatio-temporal databases. In Section 3 we discuss our approach for indexing and retrieving spatio-temporal databases. We also present an analysis of our work and a comparison with the Po-Tree [6,7]. Finally in Section 4, we discuss some assessments that we have concluded from our work.

2 Related Work

In this section we present work done in the area of spatial and spatio-temporal data indexing. We first present R-Tree [4] and then we present the Po-Tree [6,7].

2.1 R-Tree

An R-tree is a hierarchical, height-balanced external memory data structure proposed by Guttman in [3]. It is a general view of the B-tree for multidimensional spaces. Multidimensional objects are represented by what is called a Minimum Bounding Rectangle (MBR), where an MBR of an object is the smallest rectangle that encloses the object and has its sides parallel to the axes.

Figures 1 shows an example of a spatial object enclosed in a rectangle (MBR), Figure 2(a) shows a group of MBRs, and Figure 2(b) show how these MBRs are indexed in an R-tree structure.

The R-tree [4] consists of directory and leaf (data) nodes, each one corresponding to one disk page (secondary storage). Directory nodes contain entries of the form (container, ptr) where "ptr" j is a pointer to a successor node in the next level of the

Fig. 1. An MBR enclosing geometry

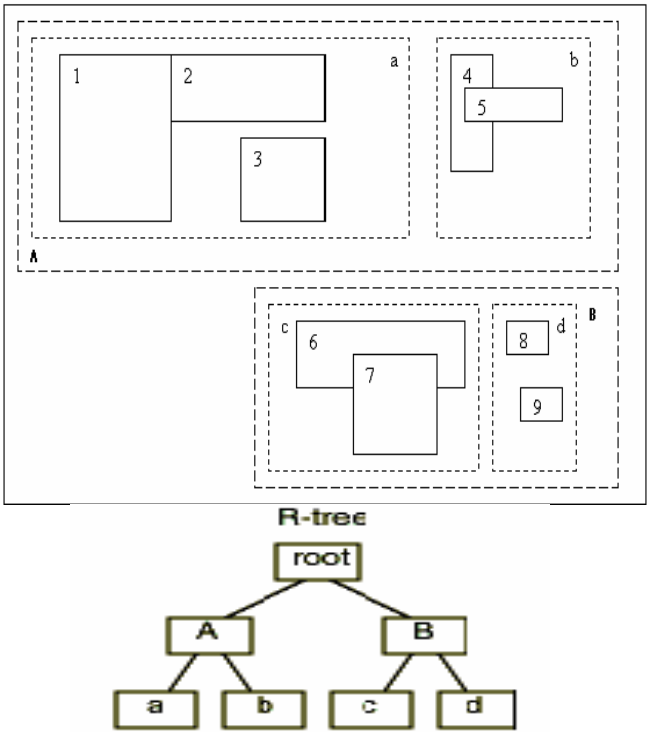


Fig. 2. (a) A collection of spatial objects in MBRs and (b) R-tree of MBRs

tree, and "container" is the MBR of all the entries in the descendant node. Leaf nodes contain entries of the form (container, oid), where "oid" is an object-identifier used as a pointer to the real object, and it is the MBR of the corresponding object. Each page can hold up to "B" entries and all the nodes except the root must have at least "m" records (usually m=B/2). Thus the height of the tree is at most $\log_m N$ where N is the total number of objects.

Searching an R-tree is similar to searching a B-tree. At each directory node, we test all entries against the query, and then we visit all child nodes that satisfy the query. However, MBRs in a node are allowed to overlap. This is a potential problem with the R-tree, since, unlike B-trees, we may have to follow multiple paths when answering a query, although some of the paths may not contribute to the answer at all. In the worst case we may have to visit all leaf nodes.

2.2 PoTree

This section presents recent work [6, 7] in spatial and temporal information structuring for natural (volcano) risk monitoring. In these papers, Noel, Servigne, and Laurini discuss methods and approaches that they designed and applied for volcanic activity monitoring of a Mexican volcano. The project uses heterogeneous types of sensors that are responsible for delivering real-time spatial information. The papers describe the following main points:

- Sensor data properties: multidimensional data (temporal and spatial) delivered by heterogeneous types of sensors.
- Volcanic activity monitoring: the use of sensors to deliver the change of spatial data of a Mexican volcano in real time, and the use of this data in the analysis of volcanic activities.
- An efficient approach for structuring the spatio-temporal data: use of an efficient structure (Po-Tree) used to store and retrieve the spatio-temporal data.

The PoTree is a spatio-temporal index with two kinds of structures: a Kd-tree and a B+ tree. The Kd-tree (a well-known structure for efficient access of spatial objects) is not a traditional tree. Instead, each leaf node in the tree points to a B+ tree. The B+ tree (also a well-known structure) uses time to order its nodes, where each node is a spatial object associated with time. The way queries are done in the Po-Tree is as follows: first the kd-tree is searched according to spatial attributes of the object then the B+-tree is also searched according the time attribute. Figure 3 represents the Po-Tree structure.

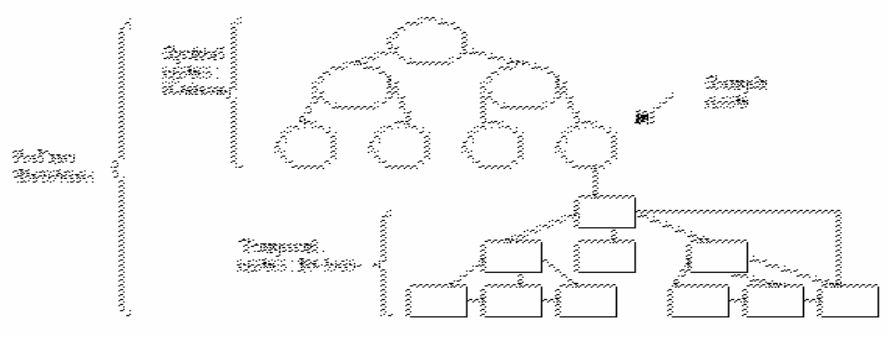


Fig. 3. The Po-Tree

3 AIRSTD Approach

In this section we introduce our approach for indexing spatio-temporal data. The approach consists of two parts: Temporal Index and spatial index. The temporal index uses the idea of the "Changes Temporal Index" [4] and the R-tree [3] is used for spatial indexing.

3.1 Temporal Index: Changes Temporal Index

In his paper [4], Halaoui presents the idea of "Changes Temporal Index" (CTI). In this paper we will extend this idea by adding spatial indexing. The final structure will be able to index spatio-temporal data according to temporal and spatial attributes as well. In this section we will briefly present what was developed in [4].

The indexing approach is as follows. Multiple versions are used to store the spatio-temporal database. Only the current version of the database contains information about all objects (or pointers to them). All old versions include objects (or pointers to them) that are valid at that version but not in the current versions. In other words, these objects

are the changes of the spatial information with respect to the current version. Figure 4 explains this idea, where the last version (current) includes the whole database that is available between some time n and now. All previous versions include only the tuples that are different at the time of their interval. We will call our structure the "Changes Temporal Index" (CTI). Note that, for better space complexity, versions may contain only pointers to the actual objects unlike what is shown in the figures.

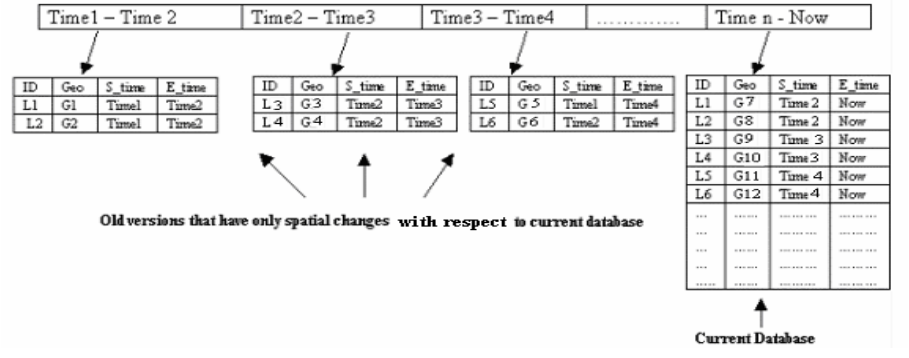


Fig. 4. Time interval index pointing to spatial changes and the current database

Moreover, Halaoui [4] introduces some rules and algorithms for manipulating and querying the CTI structure. The following is a list of all the proposed algorithms:

- Updating tuples in the current database.
- Adding a new tuple to the current database.
- Deleting a tuple from the current database.
- Querying, accessing, or building a snapshot of an old version.
- Moving to a new version

In this paper, we will consider the Binary search (could be a B-tree also) tree structure for the CTI. The reason for applying this structure is that the index is incrementally ordered according to time and it will become more efficient if we use the

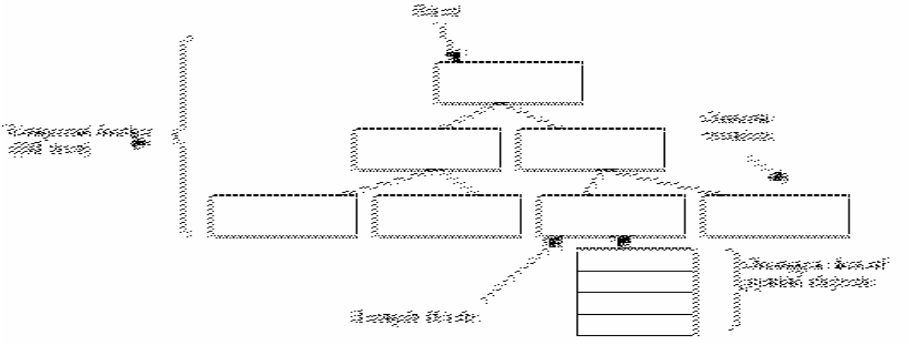


Fig. 5. Temporal index using an updated binary search tree structure

BS-trees or the B-trees. The index will look like the tree in figure 5 where there are two pointers: (1) to the root and (2) to the current version since it is more often accessed. If we apply the BS-tree, each node will be pointing to a changes version. However, if we apply the B-tree only the leaf nodes will be pointing to the changes versions.

3.2 AIRSTD Structure

AIRSTD structure is a combination of CTI and R-Tree (Figures 5 and 2). All algorithms applied for CTI and R-tree can be applied to AIRSTD. Figure 6 illustrates the AIRSTD structure that is composed of the CTI (Figure 5) and R-tree. Only the "Now" version is a full version where others point only to a list of changes with respect to the current node.

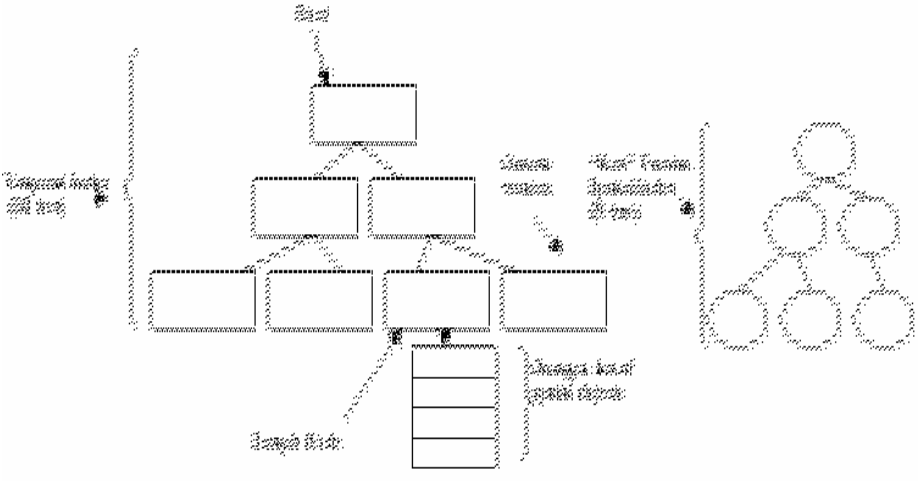


Fig. 6. AIRSTD structure

3.3 Algorithms

We have implemented and analyzed many algorithms using the proposed structure. In this section we present a list of the algorithms we have tested. However, we only present an algorithm for building a snapshot at any time T.

We have implemented and tested the following algorithms that work of the AIRSTD structure:

- Updating tuples in the current database.
- Adding a new tuple to the current database.
- Deleting a tuple from the current database.
- Searching any version at any time.
- Building a snapshot of any version at any time.

The following is the algorithm for building a snapshot at any time T (shown in figure 7):

Algorithm: Build a snapshot at time T
Input: Time "T"
Task: Return the database snapshot available at that time.

```
Step1: Traverse the current version (R-tree), return
all objects (with their time intervals) that satisfy T,
and store these objects in a temporary structure
(snapshot).
  Step2: Search the temporal index for the interval (TI)
that satisfies T.
  Step3: Go to the Tree pointed to by TI and store all
its contents in the temporary table (snapshot).
  Step4: Return the temporary table that contains the
snapshot at T.
```

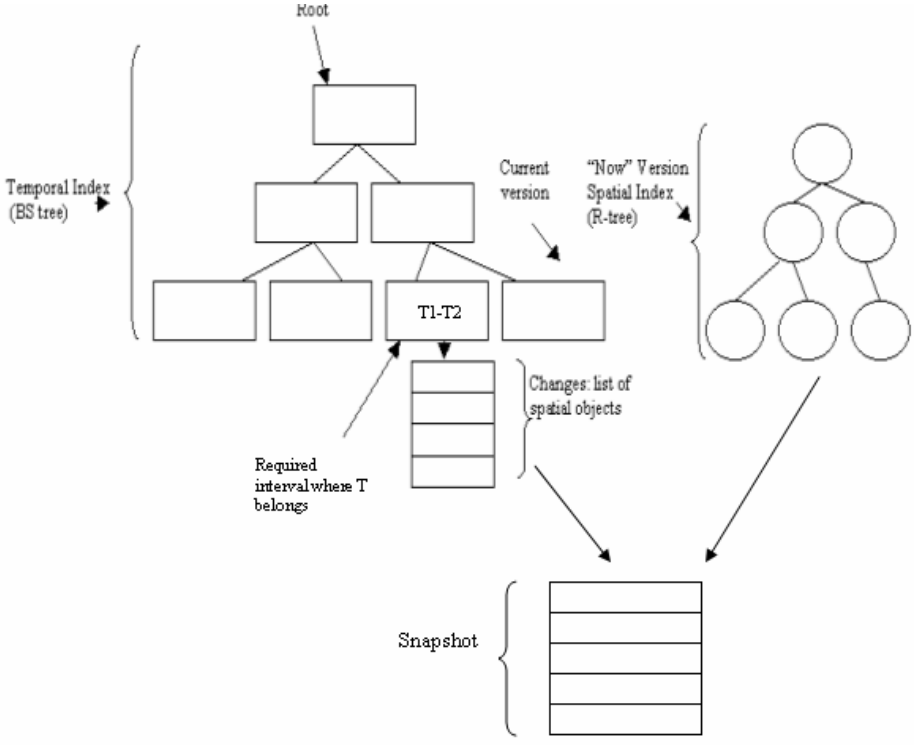


Fig. 7. Building a snapshot

This algorithm requires the following tasks:

1. Get all objects in the spatial tree
2. Search the temporal index for the interval that satisfies T.
3. Update any object from task 1 which does not satisfy T using objects from task 2.

In the worst case, task 1 requires n steps, task 2 requires log(i) steps, and task 3 requires x steps (where i is the number of intervals, x is the number of objects in the required interval, and n is the number of objects in the spatial index (R-tree)). As a result, the time complexity of the whole process is O(log(i) + n + x).

3.4 Analysis and Comparison

In this section we present a theoretical time complexity comparison between the Po-Tree and AIRSTD structure. We compare the time complexities of most used algorithms. After that we analyze this comparison and present our point of view regarding the use of each approach. Table 1 shows the time complexities of most used functionalities in the AIRSTD and Po-Tree approaches.

Table 1. Time complexity comparison table of AIRSTD and Po-Tree

Structure	Finding an object at Time T Best Case	Finding an object at Time T Worst Case	Updating or deleting an object	Building a snapshot at a Time T
PoTree	O(log(n))	O(log(n)+log(m))	O(log(n))	O(n*log(m))
AIRSTD Structure	O(log(i) + x)	O(log(i)+x+log(n))	O(log(n) + i)	O (n+ log(i) + x)

Where n is the number of objects valid at current time (valid for both structures), m is the average number of objects in each node in the Kd tree (in the PoTree), i is the number of intervals in the CTI (in the AIRSTD), and x is the average number of objects in any list that contain only changes.

If we analyze the time complexities in Table 3.1 we see that the real difference appears in two functionalities: (1) updating an object and (2) building a snapshot. The Po-Tree structure is much better when updating while the AIRSTD is better when building a snapshot. These issues can lead to the following conclusions:

1. The Po-Tree approach is better when having real time applications where updating is the most used functionality
2. The AIRSTD approach is better when having stationary application when building snapshots is required.

3.5 Experimental Results

Section 3.3 was presenting a theoretical study of the proposed algorithms and a comparison with Po-tree. In this section, we present the outcome of the experimental results gathered from applying both structures and algorithms on spatial data that is changing with time. Table 2 illustrates the experimental results of applying both AIRSTD and PoTree on one group (25000 tuple) of data with three different criterions (different rate of change, periods, and total period). The following is a brief description of the tested data:

Table 2. Experimental Results

Structure/Case	Execution Time Units Rate of change = 10 objects per 1 month Total Period = 25 years	Execution Time Units Rate of change = 0.5% per 3 months Total period = 25 years	Execution Time Units Rate of change = 20% per 1 day Total period = 2 years
Searching PoTree/Best Case	15	12	14
Searching AIRSTD/Best Case	7	134	354
Searching PoTree/Worst Case	18	13	20
Searching AIRSTD/Worst Case	20	301	4201
PoTree/Snapshot	53,201	74,003	220,012
AIRSTD/Snapshot	25,008	25,031	29,007
PoTree/Update	7	8	9
AIRSTD/Update	173	92	233

1. Discretely changing application with the following information:
    a. Interval = 3 months
    b. Total time = 25 years
    c. Rate of change = 0.5% per interval
2. Real time applications with the following information:
    a. Interval = 1 day
    b. Total time = 2 years
    c. Rate of change = 20% per interval
3. Moreover, we also tested the application with a constant number of changes (average of 10 changes per 1 month)

The results in table 2 reflect the theoretical estimations presented in table 1

4 Conclusion

We believe that most queries in spatio-temporal databases are spatial, temporal or spatio-temporal. For this reason, in this paper, we present an approach for indexing spatio-temporal data using their spatial and/or temporal information. The approach, named AIRSTD, is based on two kinds of indexing: (1) temporal using the CTI structure and (2) spatial using the R-Tree. Moreover, we present a sample algorithm for building a snapshot of using the AIRSTD structure. Finally we present a comparison table that shows the time complexities of most used functionalities of two approaches: AIRSTD and Po-Tree. We point on positive and negative features of each approach and which applications are more suitable for each.

References

1. Abdelguerfi, M., Givaudan, J.: Advances in Spatio-Temporal R-tree Based Structures. Technical report TR012-02, Computer Science Department, University of New Orleans (2002)
2. Abdelguerfi, M., Julie, G., Kevin, S., Ladner, R.: Spatio-Temporal Data Handling: The 2-3TR-tree, a Trajectory-oriented Index Structure for Fully Evolving Valid-time Spatio-Temporal Datasets. In: Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems (2002)
3. Guttman, A.: R-tree: A Dynamic Index Structure for Spatial Searching. In: Proc. of the 1984 ACM SIGMOD International Conference on Management Data, Boston, MA, pp. 47–57 (1984)
4. Halaoui, H.: Spatio-Temporal Data Model: Data Model and Temporal Index Using Versions for Spatio-Temporal Databases. In: Proceedings of the GIS Planet 2005, Estoril, Portugal (May 2005)
5. Nascimento, M., Silva, J.: Towards Historical R-trees. In: Proc. of ACM Symposium on Applied Computing, pp. 235–240 (1998)
6. Noel, G., et al.: The Po-Tree, a Real-Time Spatio-Temporal Data Indexing Structure. In: Proc. of Development in Spatial Data Handling SDH 2004, Leicester, pp. 259–270 (August 2004)
7. Noel, G., Servigne, S., Laurini, R.: Spatial and Temporal Information Structuring for Natural Risk Monitoring. In: Proceedings of the GIS Planet 2005, Estoril, Portugal (May 2005)

8. Procopiuc, C., Agarwal, P., Har-Peled, S.: Star-tree: An Efficient Self-Adjusting Index for Moving Points. In: ALENEX (2000)
9. Saltenis, S., et al.: Indexing the Positions of Continuously Moving Objects. In: Proc. of ACM SIGMOD (2000)
10. Schreck, T., Chen, Z.: R-Tree Implementation Using Branch-Grafting Method. In: Shekhar, S., Chawla, S. (eds.) Proc. of 2000 ACM Symposium on Spatial Databases: A Tour. Prentice Hall, Upper Saddle River (2003)
11. Theodoridis, Y., Sellis, P.T.A., Manolopoulos, Y.: Specifications for Efficient Indexing in Spatio-Temporal Databases. In: Proc. SSDBM, pp. 123–132 (1998)
12. Tsotras, V.J., Kangelaris, N.: The Snapshot Index, an I/O-Optimal Access Method for Time-slice Queries. Information Systems 20(3) (1995)
13. Tsotras, V.J., Jensen, C.S., Snodgrass, R.T.: An Extensible Notation for Spatio-Temporal Index Queries. In: ACM SIGMOND Record, pp. 47–53 (March 1998)

A Bridge between Document Management System and Geographical Information System

Khaoula Mahmoudi[1] and Sami Faïz[2]

[1] Laboratoire URISA -Unité de Recherche en Imagerie Satellitaire et ses Applications, Ecole Supérieur des communications de Tunis (SUPCOM),
Cité Technologique des Communications, Route de Raoued Km 3.5-2083 El Ghazala, Ariana, Tunisie
[2] Laboratoire de Télédéction et Systèmes d'Informations à Références Spatiales (LTSIRS)
Ecole Nationale des Ingénieurs de Tunis, Campus Universitaire El Manar, Tunis, Tunisie
{khaoula.mahmoudi,sami.faiz}@insat.rnu.tn

Abstract. The power of geographic information system is to help managers make critical decisions they face daily. The ability to make sound decisions relies upon the availability of relevant information. Typically, spatial databases do not contain much information that could support the decision making process in all situations. To extend the available dataset, we propose an approach for the enrichment of geographical databases (GDB) and especially their semantic component. This enrichment is performed by providing knowledge extracted from web documents to supplement the aspatial data of the GDB. The knowledge extraction process is reached through the generation of condensed representation of the relevant information derived from a web corpus. This process is carried out in a distributed fashion that complies with the multi-agents paradigm.

Keywords: Geographic Information System, Geographic Database, Multi-Document Summarization, Multi-Agents Systems, TextTiling, Rhetorical Structure Theory, Spatial Relationships.

1 Introduction

Geographic Information System (GIS) iṣa powerful set of tools for collecting, storing, retrieving, transforming and displaying spatial data from the real world. GIS is a decision support system involving the integration of spatially referenced data in a problem-solving environment [1], [2], [3]. Geographical data is often separated into two components: spatial data and attribute (called also, thematic, semantic, aspatial, descriptive) data. Spatial data is used in the visualization and manipulation of real-world objects in a computer model, e.g. roads, buildings, crime locations. Attribute data (textual, numeric, photographic, etc.) describes these real-world objects, e.g. name, cost, size, and an image of what the object looks like.

Although the use of this myriad of information, the data to be included in a GIS depends on the availability and the scale of application. Due to this restriction, we need to provide complementary data sources. For instance, a manager who is planning

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 91–102, 2009.

to build a supermarket, needs a panorama of information concerning social and economic information about the neighborhood of the designed site, public transportation and so on. Such information is usually not gathered in the same GIS. Besides, Jack Dangermond (the President of a private GIS software company) argued that "The application of GIS is limited only by the imagination of those who use it.". Hence, it is of primary interest to provide other data sources to make these systems rich sources of information.

In this context, we propose to enhance the decision-making process by enriching the descriptive component of a GDB. This enrichment will broaden the amounts of information that can be stored and made available to GIS users and by the way increase productivity, efficiencies, and lead to better decision-making.

In fact, a review of the literature shows that database enrichment is necessary to equip the initial data stored in the GDB with additional information used for a variety of purposes. One use of this data enrichment is its application within the overall generalization process. In this context, the newly acquired information was used to provide geometrical and procedural knowledge to guide the choice of the generalization solution [4], [5]. Another stream of works share in common the trend to enrich the GIS by providing means to link the unstructured data to the already available structured thematic data stored in the GDB. These works rely on already filtered data that are gathered either into digitalized libraries [7], or geographic data modules containing data from licensed sources and the public domains [8] and news data [6].

Having the same target, we extract knowledge from on-line corpus of textual documents to enrich data stored in the GDB. What distinguish our work is that we go beyond merely providing and linking the relevant documents related to geographic entities, we instead process the documents to locate the gist of information embedded into them. This processing ranges from locating the homogenous portions of texts according to a given theme, to the condensation of the detected segments. The idea behind this processing is to alleviate the GIS user from reading the linked documents, which is time consuming. Besides, our approach differs from the existing works by exploiting the spatial relations inherent to the GIS to refine the data enrichment results.

The enrichment is accomplished by using the Text Mining technique [9], [10] more precisely Multi-Document Summarization technique [11], [12]. To obtain the complementary data in a reasonable time, we propose a distributed multi-document summarization approach. In conformity with the multi-agents paradigm [13], [14], we use a set of cooperating agents, namely: an *Interface* agent, *Geographic* agents and *Task* agents. These agents collaborate to lead jointly the system to an optimal solution: an optimal summary. The approach we propose is modular. It consists of three stages: segment and theme identification, delegation and text filtering. Besides these basic stages, a refinement stage can be eventually executed when the user is unsatisfied by the results generated by the data enrichment process.

In this paper, we report mainly the implementation of our approach presented with extensive details in [15], [16], [17] and the integration of the resulting tool to an existing open GIS. Likewise, we put the emphasis on the refinement process by exploiting the spatial relations.

The remainder of this paper is organized as follows: In section 2, we provide an overview of our data enrichment process. In section 3, we describe our tool: we detail

the major functionalities of the tool as well as the refinement process. Finally, in section 4, we report the implementation.

2 Our Data Enrichment Process

One of the characteristics of the GIS applications is their heavy requirements in data availability. In order to be successful, a GIS must integrate a seemingly endless amount of spatially related information into a single, manageable system. This remains an issue.

In this context and in an attempt to provide timely, accurate and easily accessed information, we propose to enrich the GDB embedded into GIS. Indeed, more timely and reliable information result in better decisions. Our intention is to support decision makers while making decisions that rely on the information given by a GIS, by providing other complementary data sources. To enrich data stored in the GDB, we extract knowledge from on-line corpus of textual documents. This is accomplished by using multi-document summarization [18], [19]. In fact, while the information overload problem grows, the user needs automated tools to retrieve the gist of information from the increasing amount of information. This is due to the difficulty to fully read all retrieved documents when we deal with a large sized corpus of documents.

To obtain the complementary data in a reasonable time, we propose a distributed multi-document summarization approach. This distribution is justified by the fact that the summary generation among a set of documents can be seen as a naturally distributed problem. Hence, each document is considered as an agent, working towards a concise representation of its own document which will be included as part of the summary of the whole corpus.

The society of agents, shelters three classes of agents: the *Interface* agent, the *Geographic* agents and the *Task* agents. Independently of its type, each agent enjoys a simple structure: the acquaintances (the agents it knows), a local memory gathering its knowledge, a mailbox for storing the received messages that it will process later on. The interaction among agents is performed by means of message passing. It deals with a peer-to-peer communication.

Our approach [15], [16], [17] to the data enrichment is a three-stage process_consisting of: segment and theme identification, delegation and text filtering. At the first stage, the documents of the corpus are processed in such a way that the portions of the texts that convey the same idea are depicted. These portions known as the segments are marked with their themes. The latter are the most frequent concepts. Thereafter, to each theme tackled throughout the corpus, we affect a delegate agent responsible of filtering the relative segments. This delegation is fulfilled by minimizing the work overload and the communication overhead. The ultimate stage of the overall knowledge extraction process is the text filtering. This filtering is achieved by performing a rhetorical structure analysis of the texts. It consists of eliminating the portions of texts that are not essential for the comprehension of the main idea delivered by the text at hand. The retained text portions are included to the final summary.

In fact, once these three main stages are performed, the user can check the relevancy of the generated results. In the case of unsatisfaction, the user can launch a refinement process in a hope to improve the results.

To accomplish these different steps, an enrichment system was set up. In the sequel, we first detail the main stages of the system, second we present the refinement process.

3 The SDET Tool

To perform the data enrichment process outlined above, we have developed the SDET tool. SDET stands for Semantic Data Enrichment Tool.

In what follows, we first describe the main functionalities of our SDET tool. Second, we report the refinement process.

3.1 The Main Functionalities of SDET

The SDET tool was developed to enrich the descriptive data stored in the GDB. This tool consists of a set of functionalities. It deals with: first, the process launching, second, the identification of the segments and their relative themes and third, the extraction of the gist of the text.

Process Launching. The overall data enrichment process is triggered whenever the GDB data in response to the GIS user query is not satisfactory enough (lack of data, insufficient details...). In such case, the system creates an *Interface* agent. The latter, receives as input a corpus of documents relative to the geographic entity (or entities) at hand. This on-line corpus is a result of an information retrieval (IR). The retrieved web documents are to be distributed by the *Interface* agent among the *Task* agents for a content processing. These agents are created by the *Interface* agent.

We notice that whenever the user is dealing with more than one geographic entity, the *Interface* agent creates a set of *Geographic* agents each of which is responsible of one geographic entity. Hence, each *Geographic* agent governs a sub-set of *Task* agents responsible of processing the relative corpus.

In fact, the GIS user can also profit from the results of a previous IR session that are already stored on the disk.

Segment and Theme Identification. At this stage, the documents distributed among the *Task* agents are processed to delimit the portions of the texts that are thematically homogenous and to annotate them with the most significant concept, which we call its theme.

To start this stage, each *Task* agent first cleans its document in order to keep only the meaningful text. This parsing allows to discard the formatting tags and other auxiliary information. Furthermore, a stop-list containing the common words (prepositions, articles…) is given as input to filter the parsed texts and to keep only the words having a semantic content. Then, the resulting texts are stemmed [20] to reduce words to a stem which is thought to be identical for all the words linguistically and often conceptually related. For example, the words learner, learning, and learned would all map to the common root learn. After these pre-processing steps, the texts are segmented by adopting the TextTiling algorithm [21]. Hence, the text is seen as portions (segments) that are topically homogenous. For these segments, the *Task* agents look for annotating them with the most salient concepts.

To achieve this purpose, each pre-processed document is passed to a tagging program [22]. The aim is to assign a part-of-speech like noun, verb, pronoun, preposition, adverb, adjective or other lexical class marker to each word in a sentence. From all these part-of-speech tags, we keep only those identified as nouns, which are considered the more meaningful. These nouns are fed as input to the theme identification module. In fact, we compute the frequency of each noun occurrence within its segment. If one noun frequency is highest enough (above a given threshold) it is affected as the topic of the segment. Nevertheless, this is not always the case; it is frequent to have a text as a set of terms, which together contribute to develop a subject. Here, the terms are homogenously distributed and none of them is remarkably reoccurring. In front to this situation, we make use of WordNet [23]. By doing so, we intend to depict the term to which all (almost all) terms of the segment are related. Then, the new frequencies are set to the number of links that the noun maintains with the other nouns in the segment. Hence, the most referenced word via semantic relations will be assigned as the topic of the text. For the sake of clarity, we consider the following text fragment: *"The state-run Tunisian Radio and Television Establishment (ERTT) operates two national TV channels and several radio networks. Until November 2003 the state had a monopoly on radio broadcasting"*.

This fragment of text is first processed as abovementioned. From the noun frequencies we note that none of them is highest enough to deduce the topic. Hence, we explore the relations embedded into WordNet. We depict the following relations: *hypernym(TV)=broadcasting, hypernym(radio)=broadcasting, synonym(TV)=television, topic-domain (network)= broadcasting*. Then we infer that *"broadcasting"* is the noun that reveals the main idea of the text.

Thereafter, to each detected theme in the corpus, the *Interface* agent (or the concerned *Geographic* agent in the case of a set of geographic entities) affects one *Task* agent as a delegate responsible of processing the relative segments. A delegate maintains a set of *generated documents*, each of which consists of the gathering of all the segments related to the same topic. Then, each delegate has to perform a text filtering task.

Text filtering. At this stage of the enrichment process, the generated documents are under the jurisdiction of the *Task* agents known as delegates. The filtering is carried out by building the rhetorical structure trees [24], [25], [26] for the segments selected by the GIS user. The selected segments are split into lexical units (sentences or clauses).

By using a database storing the cue phrases (markers like for example, but…), the delegate agent builds an XML file reflecting the rhetorical structure of the text. This XML file is used to facilitate and guide the building of the rhetorical tree. Hence, for each detected cue phrase having some entries in the database, the equivalent leaves and the internal nodes are built. Whenever, no such cues are depicted, the tree building is performed by computing the similarity between the concerned units according to a given formula [16], [17]. According to the similarity value, we decide which unit to include in the summary.

By sweeping the resulting tree in a top-bottom fashion, we depict the units that are most important for the comprehension of the texts. For instance, if the detected cue phrase is "*for example*", the unit to which the cue belongs can be discarded because it is just an exemplification supporting a previous unit.

The units derived from each generated document form a summary of a given theme. Hence, each delegate generates a set of partial summaries to be included as a part of the final summary of the whole corpus of documents. Such information are dispatched towards the GIS user by the *Interface* agent to judge its relevancy according to his requirements.

3.2 Refinement Process

Once the knowledge embedded into the corpus of on-line documents are retrieved, the user can judge the relevancy of the results to his expectations. In fact, the purpose of our data enrichment process is to alleviate the task of GIS user in quest of information (namely finding the more relevant documents, reading a huge mass of collected documents, locating the more salient informational zones within these documents).

Sometimes, the data enrichment process is fruitless. This may be engendered from the IR step. This step can be blemished by a lack of information or a certain ambiguity; hence the decision can't be made. In the first case, we have to deduce the information from the available ones. In the case of ambiguity, especially when the same noun refers to more than one geographic entity; the GIS user is overwhelmed by a large set of non-relevant documents.

To obtain the desirable information relative to the target entities, one has to describe the latter in an accurate way. By doing so, we have more chance to retrieve the relevant documents.

In fact, because the geographic entities are not disparate rather they have spatial relationships with the other entities, one can exploit the spatial relationships [27] [28] to better locate them. The idea is to examine the vicinity of the geographic entity subject of the data enrichment in order to disclose the geographic entities that have some relations with it. Our argument is the first law of geography that describes the nature of geographic systems in which *"everything is related to everything else, but near things are more related than distant things"* [29].

To perform this refinement we exploit the following spatial relationships: adjacency (what adjoins what), proximity (how close something is to something else), connectivity (topological property of sharing a common link, such as a line connecting two points in a network) and the nearest neighbor spatial relation. Besides these relationships, we exploit the overlaying operation. Overlay analysis is the process of superimposing two or more layers, such that the resultant maps contain the data from both maps for selected features. Hence this allows to match different data layers for the same geographic area, which enables to visualize the interactions among the different data.

By considering all these spatial relations, many scenarios can occur. In the sequel some of these scenarios are reported.

Consider one scenario where the GIS user is looking for information about the contamination state of a river. If the user fails to found such information in the documents relative to the river, it will be intersecting to examine the connected rivers through a web search. The state of the connected branches may be worth informing about the state of the river subject of the query. Hence, we enlarge our investigation space to look for information through documents relative to the related entities according to the connectivity topological relation. By the way, the GIS user can make decision

about the state of the river or the measurement to undertake in order to prevent eventual contamination.

Another spatial use case, is to profit from the overlay analysis. Assume that a GIS user is handling a GDB relative to birds' dispersion all over the world, and that our data enrichment process is not satisfactory enough for one of the reasons mentioned above. The user can refine the results by overlaying the layer of birds' dispersion and the countries to define the area where both inputs overlap and retains a set of attribute for each. The points of intersection are the keys of the search that are likely to return the relevant documents. In other words, the couple of identifiers relative to the bird and the country contribute as the components of the key used for the IR that contribute to better locate the entity and can enhance the results.

The ambiguity is another problem that confront the user during the enrichment process. One illustration of the ambiguity is a name, which can refer to more than one geographic entity, this is the case with the name Carthage. We notice that the results of the IR session about Carthage (here we are looking for the historical city of Tunisia) returns about 8670000 documents. These results are huge to be handled in a reasonable time and a large subset of documents from the resulting corpus is not relative to our geographic entity. In fact, many documents are not relevant to Carthage of Tunisia, for instance, some documents are relative to the Carthage city located to U.S.A. (www.carthagetx.com). To deal with this situation, we can exploit one of the spatial relations to describe the geographic vicinity of the city. The references of the detected geographic entities are to be used to enhance and refine the results, and this by restraining and reaching the documents that correspond really to the target entity.

4 Implementation

The SDET functionalities have been implemented and integrated into an open GIS. The implementation was performed using the Java programming language. Indeed, to comply with our distributed architecture, we have used Java which supports the multithreaded programming. A multithreaded program contains two or more parts that can run concurrently.

Concerning the GIS platform, and from a comparative study of the existing open GIS, we have opted for the use of the open JUMP (Java Unified Mapping Platform) [30], [31]. This choice relies on the fact that open JUMP GIS provides a highly extensible framework for the development and execution of custom spatial data processing applications. It includes many functions common to other popular GIS products for the analysis and manipulation of geospatial data.

In what follows, we report some screenshots showing the major capabilities of our system.

The enrichment process is launched once a GIS user whose is looking for information about the geographic entities is not satisfied. As shown in Fig. 1, the user has firstly to choose the geographic entity (or entities). In our case, we deal with a Shape File format relative to the geographic entities: The world map. The figure shows the results of querying the GDB about the entity Tunisia.

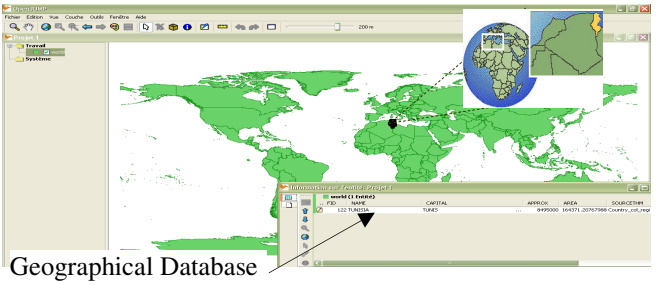


Fig. 1. The GDB enquiring relative to the geographic entity: country

Here the user is unsatisfied, he resorts to enrich the results by adding complementary information. The Fig. 2, shows the launching of the enrichment process. The user launches an IR session carried out by using the Google SOAP Search API [32], [33]. This search can be generic or depending on some topics specified by a set of keys. In fact, the user can use a corpus resulting from an anterior IR session. The retrieved corpus is to be processed by our society of agents.

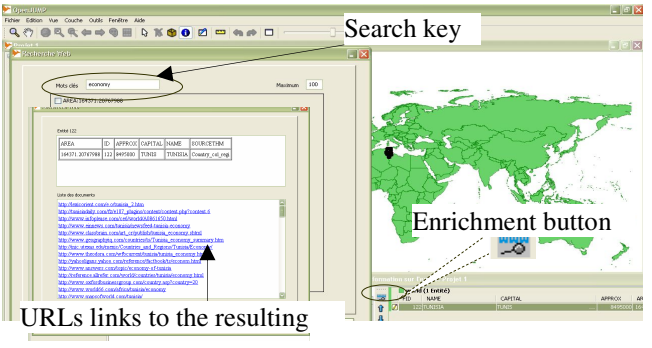


Fig. 2. Launching of the data enrichment process

Because the architecture of our system is modular, we have reported the results of each step to maintain a certain interactivity with the GIS user. The results of the segment and theme identification are shown in Fig. 3. This figure shows the detected themes that are stored in a list. This allows to the user to navigate among the generated documents relative to these themes. Besides, we maintain mappings to the original documents. This is useful to consult the integrity of the original web document if we judge that the relative segments are pertinent. Moreover, the user can store the generated documents.

To keep only the most salient portions of text from the generated documents, we have proposed three options to the GIS user (see Fig. 4). He can condensate one segment, the whole generated document, or the whole generated document with the possibility to discard the segments that are likely to report the same information. This is accomplished by computing the similarity (according to a given threshold) between different segments of a generated document. In effect, whatever the user choice, in

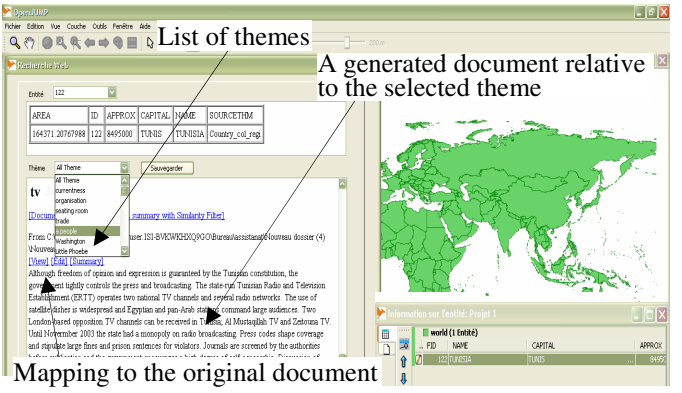


Fig. 3. The segment and theme identification

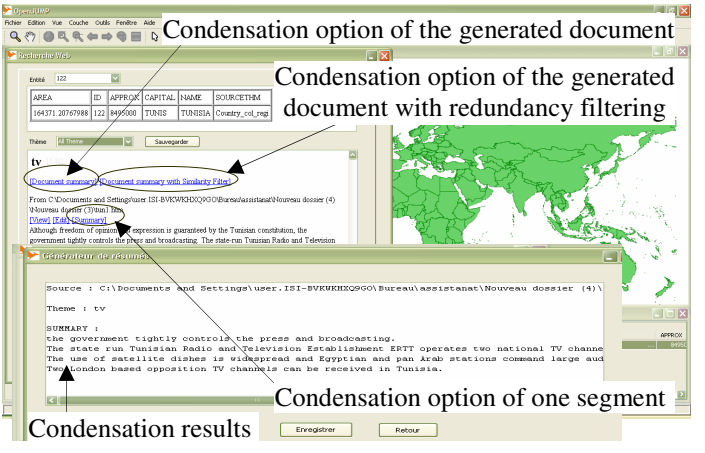


Fig. 4. The text filtering

back-office the system agents generate rhetorical trees of the selected texts (see Fig. 5). The results can be stored for a future consultation.

Once the results of the data enrichment are not satisfactory enough, the user can launch a data refinement process to enhance the results. One example is shown in Fig. 6 relative to the exploitation of the adjacency relation. In this situation, a user is looking for information about Uganda and the relationships maintained with its neighbors that he ignores. Such information is not stored in our GDB relative to the world countries. By exploiting the adjacency relation, the system highlights graphically the neighboring countries meanwhile provides the attributes that will serve as part of the search key. Besides, to guide more the GIS user, we report the statistics relative to the number of web documents reporting information of the association of each neighbor with Uganda. Hence, the user can be adverted about the amount of data to handle and by the way assist him in choosing the attributes of the geographic entities that are likely to allow reaching the relevant documents.

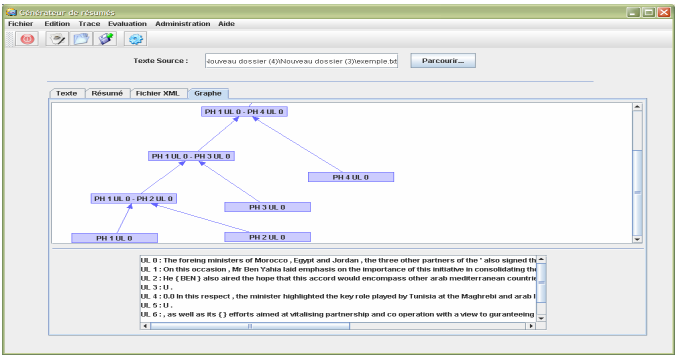


Fig 5. A rhetorical tree

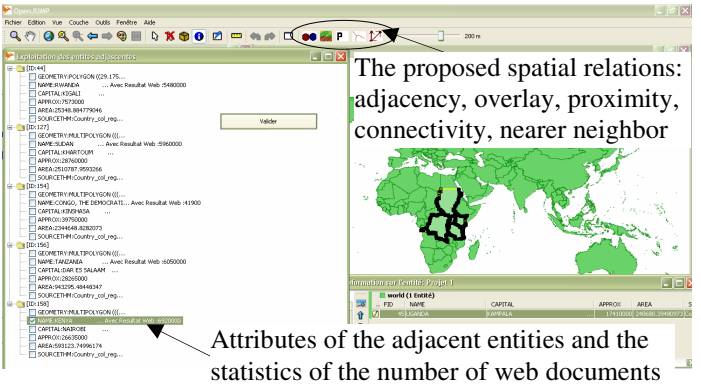


Fig. 6. Refinement of the enrichment process via the adjacency relation

5 Conclusion

With the increasing demand for the geographical information expressed by the GIS users, the data enrichment processes play a key role to extend the dataset already stored within the system.

In this context, we propose an approach to provide complementary data that enrich the descriptive aspect of the geographic databases.

This paper describes a semantic data enrichment tool that we call SDET (Semantic Data Enrichment Tool). The overall enrichment process was presented. Then, the main SDET tool functionalities as well as their integration to an open GIS were detailed. Furthermore, a refinement process was reported. It is provided to GIS users as a mean to describe with more accuracy the geographic entities and by the way to increase the chance to reach the pertinent documents. This is achieved by exploiting the spatial relations inherent to the GIS.

References

1. Faïz, S.: Geographic Information System: Quality Information and Data mining. Edn C.L.E (1999)
2. Brian, E.M.: Understanding the Role of Geographic Information Technologies in Business: Applications and Research Directions. Journal of Geographic Information and Decision Analysis 1(1), 44–68 (1997)
3. Rigaux, P., Scholl, M., Voisard, A.: Spatial Data Bases. Morgan Kaufmann, San Francisco (2001)
4. Plazanet, C.: Geographical database enrichment: analysing the geometry of linear features for automated generalization (application on roads). Ph.D. thesis, IGN, France (1996)
5. Neun, M., Weibel, R., Burghardt, D.: Data Enrichment for Adaptive Generalisation. In: ICA Workshop on Generalisation and Multiple representation, Leicester (2004)
6. Hyland, R., Clifton, C., Holland, R.: GeoNODE: Visualizing News in Geospatial Environments. In: Proceedings of the Federal Data Mining Symposium and Exhibition 1999, AFCEA, Washington D.C (1999)
7. David, A.S.: Detecting events with date and place information in unstructured text. In: Proceedings of the 2nd ACM+IEEE Joint Conference on Digital Libraries, Portland, OR, pp. 191–196 (2002)
8. MetaCarta, G.T.S.: versus MetaCarta GeoTagger. MetaCarta, Inc. (2005)
9. Tan, A.: Text Mining: the State of the Art and the Challenges. In: Zhong, N., Zhou, L. (eds.) PAKDD 1999. LNCS, vol. 1574, pp. 71–76. Springer, Heidelberg (1999)
10. Weiss, S., Apte, C., Damerau, F.: Maximizing Text-Mining Performance. IEEE, Intelligent Systems 14(4) (1999)
11. Goldstein, J., Mittal, V., Carbonell, J., Kantrowitz, M.: Multi-document summarization by sentence extraction. In: Proceedings of the ANLP/NAACL-2000 Workshop on Automatic Summarization, Seattle, WA, pp. 40–48 (2000)
12. Harabagiu, S., Finley, L.: Generating single and multi document summaries with GISTEXTER. In: Proceedings of the Workshop on Multi-Document Summarization Evaluation of the 2nd Document Understanding Conference at the 4Oth Meeting of the Association for Computational Linguistics, Philadelphia (2002)
13. Ferber, J.: Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence, 1st edn. Addison-Wesley Professional, Reading (1999)
14. Wooldridge, M.: Introduction to MultiAgent Systems, 1st edn. John Wiley & Sons, Chichester (2002)
15. Faïz, S., Mahmoudi, K.: Semantic enrichment of geographical databases, in. Encyclopedia of database technologies and applications. Edn. Idea Group, Etats-Unis, pp. 587–592 (2005)
16. Mahmoudi, K., Faïz, S.: Une approche distribuée pour l'extraction de connaissances: Application à l'enrichissement de l'aspect factuel des BDG. Revue des Nouvelles Technologies de l'Information. Edn. Cépaduès, 107–118 (2006)
17. Mahmoudi, K., Faïz, S.: L'apport de l'information spatiale pour l'enrichissement des bases de données. In: INFORSID 2006, Hammamet Tunisie, pp. 323–338 (2006)
18. Radev, D.R., Jing, H., Budzikowska, M.: Centroid-based Summarization of multiple Documents: sentence extraction, utility-based evaluation, and user Studies. In: ANLP/NAACL Workshop on Automatic Summarization, Seattle, pp. 21–19 (2000)
19. McKeown, K., Klavens, J., Hatzivassiloglou, V., Barzilay, R., Eskin, E.: Towards Multidocument Summarization by Reformulation: Progress and Prospects. In: AAAI/IAAA (1999)

20. PorterStemmer (2005), http://www.tartarus.org/~martin/PorterStemmer/
21. Hearst, M.A.: TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages. Computational linguistics 23, 33–46 (1997)
22. http://web.media.mit.edu/~hugo/montytagger (2006)
23. Miller, G.: WordNet: An on-line lexical database. International Journal of Lexicography (special issue) 3(4), 235–312 (1990)
24. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: Toward a functional theory of text organization. An Interdisciplinary Journal for the Study of Text, 243–281 (1988)
25. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: A theory of text organization. Technical Report ISI/RS-87-190, USC, Information Sciences Institute (1987)
26. Marcu, D.: Discourse Trees are good Indicators of Importance in Text. In: Mani, Maybury (eds.) Advances in Automatic Text Summarization, pp. 123–136. MIT Press, Cambridge (1999)
27. Clementini, E., Sharma, J., Egenhofer, M.J.: Modeling topological spatial relations: strategies for query processing. Computers and Graphics, 815–822 (1994)
28. Egenhofer, M.J.: Query processing in spatial-query by sketch. Journal of visual languages and computing 8(4), 403–424 (1997)
29. Bin, J., Itzhak, O.: Spatial Topology and its Structural Analysis based on the Concept of Simplicial Complex. In: 9th AGILE Conference on Geographic Information Science, Visegrád, Hungary, pp. 204–212 (2006)
30. Opengeospatial (2006), http://www.opengeospatial.com
31. Open jump (2006), http://www.jump-project.org/
32. http://www.google.com/apis/ (2006)
33. http://actualites.webrankexpert.com/200304-google-api.htm (2006)

A New Digital Notary System*

Kaouthar Blibech and Alban Gabillon

LIUPPA/CSySEC,
Université de Pau – IUT de Mont de Marsan
371 rue du ruisseau, 40000 Mont de marsan, France
k.blibech@etud.univ-pau.fr, alban.gabillon@univ-pau.fr

Abstract. Timestamping is a cryptographic technique providing us with a proof of existence of a message/document at a given time. Several timestamping schemes have already been proposed. In this paper, we shortly review existing schemes and then fully define a new timestamping system based on skip lists. We show that our scheme offers good performances.

Keywords: Timestamping, totally ordered, partially ordered, authenticated dictionaries, skip lists, security analysis, performances.

1 Introduction

Timestamping is a technique for providing proof-of-existence of a message/document at a given time. Timestamping is mandatory in many domains like patent submissions, electronic votes or electronic commerce. Timestamping can ensure non-repudiation. Indeed, a digital signature is only legally binding if it was made when the user's certificate was still valid, and a timestamp on a signature can prove this. Parties of a timestamping system (also called digital notary system) are the followings:

- **Client:** forms the *timestamping request* which is the *digest* of the document to be timestamped. The client computes this digest by using a well known one way[1] collision-free[2] hashing function. Submitting the digest of the document instead of the document itself preserves the confidentiality of the document.
- **TimeStamping Authority** (TSA)**:** receives the timestamping request at time *t* and issues the *timestamp*. The timestamp is a proof that the digest was received at time *t*. The TSA produces the timestamp according to a *timestamping scheme*.
- **Verifier:** verifies the correctness of the timestamp by using the *verification scheme* corresponding to the timestamping scheme which was used to produce the timestamp.

Most of the existing timestamping schemes are *linking* schemes. Linking schemes were introduced by Haber and Stornetta [14]. Such schemes significantly reduce the scope of operations the TSA has to be trusted for. Basically, they work as follows:

* This work was supported by the Conseil Général des Landes and the French ministry for research under ACI Sécurité Informatique 2003-2006, Projet CHRONOS.

[1] One-way means that no portion of the original document can be reconstructed from the digest.

[2] Collision-free means that it is infeasible to find x and x' satisfying $h(x) = h(x')$.

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 103–114, 2009.

During a time interval which is called a *round*, the TSA,

- receives a set of timestamping requests,
- aggregates the requests in order to produce a *round token*,
- returns the timestamps to the clients. Each timestamp consists of the round token, the digest and the authentication path proving that the round token depends on the digest.

Each round token is one-way dependent on the round tokens issued before. Round tokens are regularly published in a widely distributed media (a newspaper). After the publication it becomes impossible to forge timestamps (either to issue fake ones afterwards, or modify already issued ones), even for the TSA.

In the case of *partially ordered linking schemes* [2][3][4], only timestamps from different rounds are comparable whereas in the case of *totally ordered linking schemes* [11][10][14], the temporal order of any two timestamps can be verified even if these two timestamps belong to the same round. Partially ordered schemes are generally simpler than totally ordered schemes. However, as mentioned by Arne et al. [1], since totally ordered linking schemes allow us to compare two timestamps of the same round, longer rounds can be used. Using longer rounds enables reducing the amount of data to be published and the amount of data to be verified.

The purpose of this paper is to define a new totally ordered scheme which is simpler than the existing ones and which shows optimal performances. Our scheme uses a skip list. A skip list is a data structure which was defined by Pugh [21]. This paper is organized as follows. Section 2 reviews related works. Section 3 presents our scheme. Section 4 deals with performance issues. Finally section 5 concludes this paper.

2 Survey of Timestamping Systems

2.1 Partially Ordered Schemes

In those schemes, the authority aggregates a set of timestamping requests in order to produce a round value. Round values are regularly published. Requests of the same round can not be ordered while the temporal order of any two requests of different rounds can be verified.

Merkle Tree scheme. Most of the timestamping protocols use a binary tree structure also called *Merkle Tree* [19][20]. However, this method is not always accurate. This is trivially the case when the number of timestamped documents is very small while the frequency of publication is very low. In that case, the accuracy of the timestamp may not satisfy the client. Notice also that this method is not practical when the number of documents is not close to a power of 2. In fact, if it is the case, then we need to add a number of padding elements to make it a power of 2.

Accumulator scheme. Accumulators where introduced by Benaloh and de Mare in 1993 [4]. By accumulators they designate a family of function having some properties[3]. Modular exponentiation is a typical accumulator. A timestamping authority can use the modular exponentiation to compute a round token by accumulating the

[3] Describing these properties is out of the scope of this paper.

received requests. The main drawback of this scheme is that it depends on an RSA modulus n [4]. Indeed, if the authority would know the factorization of the RSA modulus then she would be able to compute fake timestamps. Therefore, this method introduces a practical problem: who should be the trusted entity computing the RSA modulus? Notice also that the modular exponentiation is slower than simple hashing operations.

2.2 Totally Ordered Schemes

In those schemes, every timestamping request is linked to the other requests in such a way that requests of the same round can be ordered. The authority produces round values that are published.

Linear Linking schemes. The first totally ordered scheme which was defined in [14] links the timestamping requests in a linear chronological chain. By this mean, the authority creates a linear cryptographic dependency between successive requests. Some values of the resulting chain are then published and provide us with an absolute time. The verification process consists in recomputing the entire chronological chain between the publications that bracket the timestamping request. Verification is then costly. It is the main drawback of this scheme.

Binary linking schemes. In order to provide relative ordering, Buldas and al. use a simply connected authentication graph for their timestamping scheme [9]. Buldas and al. proved that there is a verification chain between each pair of elements so that the relative order of timestamped requests can be proved. However, their scheme is less efficient than Merkle tree scheme. This is due to the additional concatenation operations both for construction and verification. In addition, time-stamping tokens are greater than the tokens produced by Merle Tree Scheme. This scheme is also more difficult to understand and to implement than the previous schemes. Finally, it has the same constraint on the number of inserted elements than the Merkle tree scheme.

Threaded tree schemes. In order to improve the efficiency of the previous scheme, Buldas and al. developed a new scheme [10] based on Merkle tree structure. This scheme is easier to implement than the binary tree scheme and provides smaller time-stamping tokens. But, when compared to the Merkle tree scheme, this scheme has larger time complexity, both for construction and verification, due to the additional concatenation operations. Moreover, this method has the same constraint on the number of inserted elements than the binary linking scheme or the Merkle tree scheme.

3 A New Timestamping Scheme

3.1 Skip Lists

W. Pugh introduced skip lists as an alternative data structure to search trees [21]. The main idea is to add pointers to a simple linked list in order to skip a large part of the list when searching for a particular element. While each element in a simple linked list points only to its immediate successor, elements in a skip list can point to several successors. Skip lists can be seen as a set of linked lists, one list per level (see figure 1).

All the elements are stored at the first level 0. A selection of elements of level k is included in the list at level k+1. In *perfect* skip lists (see figure 1), which are the most known skip lists, element e belongs to level i if its index is a multiple of 2^i. Consequently, element at index 5 belongs only to the first level, while element at index 4 belongs to the three first levels. In figure 1, B and E nodes are stored at all levels and called *sentinel* elements. The highest B node is called *starting node St*. The highest E node is called *ending node Et*. In the example of figure 1, nodes contain the elements of the set {5,10,13,14,15,35,34}. Edges are the pointers. Numbers *[0..3]* are the levels. Numbers *[1..7]* are the indexes.

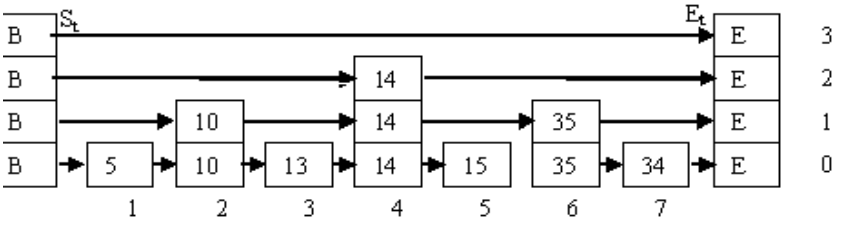


Fig. 1. An example of skip list

3.2 Timestamping Scheme

In [5], we defined an *authenticated dictionary* based on skip lists. An authenticated dictionary is a data structure that supports both update queries and *tamper-evident* membership queries. A tamper-evident membership query is of the form "does element e belong to set S?". If e belongs to S then the answer to such a query is a proof that e belongs to S. In [6], we sketched a new totally linking timestamping system based on the authenticated dictionary we defined in [5]. The purpose of this paper is to fully define this new timestamping system and to study its performances.

Our scheme uses one *append-only* perfect skip list per round. Elements of the skip lists are the timestamping requests. Each new request is appended to the skip list. Since we are dealing with perfect skip lists, each element of the skip list is associated to one or several nodes according to the index of the request. Each node has the following four properties:

- its *value*, which is a timestamping request (digest)
- its *level*, ranging from 0 to the highest level of the skip list
- its *index*, which is its position in the skip list
- its *label*, which is a hash value one way dependent on the labels of the previous nodes.

Nodes associated to the same element have the same value and index. For example, let us consider nodes a and p in figure 2. They have the same index (20) and value (h_{20}). Level of node a is 2 whereas level of node p is 0. Labels are not shown but are different. The label of the starting node is the round token of the previous round whereas its value is the last request which was received during the previous round. Basically, our scheme works as follows:

- Alice sends a timestamping request which is the digest h of a document.
- The TSA appends h to the skip list and immediately returns to Alice a signed *acknowledgment* containing the index of h in the skip list and the proof that h is inserted after the elements which are already in the skip list. We call this proof the *head proof* (see algorithm 1).
- The TSA computes the label of each node associated to element h (see algorithm 2).
- At the end of the round, the TSA inserts the last request which becomes the ending sentinel element. The label of the ending node is the round token.
- The TSA publishes the round token and sends to Alice (and other clients) some additional information allowing her to prove that her request belongs to the round whose token has been published. We call this information the *tail proof* (see algorithm 3). The final timestamp consists of the digest h, the index of h, the head proof, the tail proof and the round token.
- If a verifier, Bob, needs to check the validity of the timestamp then he has to verify that he can compute the round token from h, the index of h, the head proof and the tail proof. Bob does the verification by processing algorithm 4.

Figure 2 shows the insertion of h_{21} at index 21. h_{16} to h_{21} are requests (digests of documents). Numbers [16..21] are indexes. Labels are not shown. The arrows denote the flow of information for computing the labels (see algorithm 2). The head proof for h_{21} consists of the labels of the dark grey nodes (nodes *q, o* and *a*) (see algorithm 1).

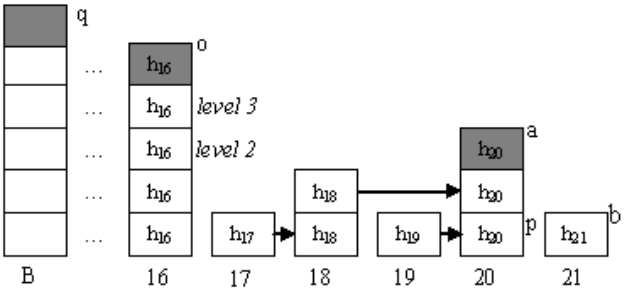


Fig. 2. Insertion of h_{21}

Figure 3 shows the insertion of the ending node (last request of the round). The arrows denote the flow of information for computing the labels (see algorithm 2). The label of the ending node is the round token. The tail proof for h_{21} consists of the value h_{22} and the labels of the light grey nodes (nodes *r* and *x*) (see algorithm 3). Note that the last request of the round is h_{25}. Since it is the ending element, it belongs to all levels although 25 is not a multiple of 2^5. Figure 3 shows also the verification process for h_{21}. Labels of thick nodes are computed during the verification process (see the next section 2.3 about verification).

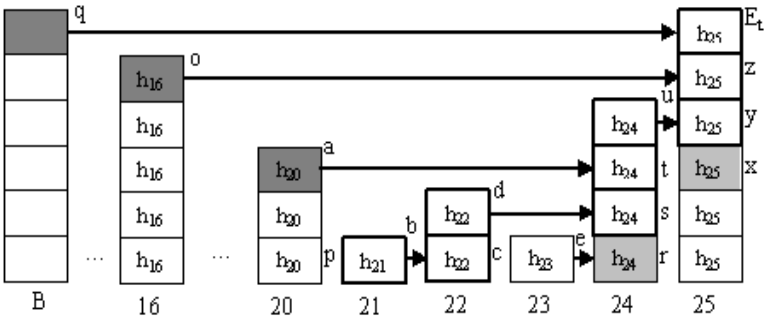


Fig. 3. Insertion of the ending element

```
Algorithm 1. Head Proof Computation
1: hp := {}
2: For i ranging from 0 to height(S)
3:    If last(i) is a plateau node then
4:         append label(last(i)) to hp
```

```
Algorithm 2. Hashing Scheme
1:If down(n) = null, {n is at level 0}
2:  If left(n) is not a plateau node then
3:     label(n):= value(n)
4:  Else
5:     label(n):= hash(value(n)||label(left(n)))
6:Else
7:  If left(n) is not a plateau node then
8:     label (n):= label(down(n))
9:  Else
10:    label(n):= hash(label(down(n))||label(left(n)))
```

Algorithm 1 is used to compute the head proof (*hp*) of the newly inserted element *h*. *S* denotes the skip list. Function *height(S)* returns the highest level of the skip list *S*. Function *last(i)* returns the last node which was inserted at level *i* before the insertion of *h*. Function *label(n)* returns the label of node n. We define the *plateau* node of an element as the highest node associated to that element. Let us consider the nodes having an index strictly lower than the index of *h*. Among these nodes, for each level *l*, let us consider the node which has the greatest index. If it is a plateau node then its label belongs to the head proof. Figure 3 shows that the head proof of index 21 consists of the label of node *a* (that will be used during the verification to compute the label of node *t*), the label of node *o* (that will be used during the verification to compute the label of node *z*) and the label of node *q* (that will be used during the verification to compute the label of the ending node i.e. the round token).

Algorithm 2 is used to compute the labels of the nodes associated to the newly inserted element *h*. Function *value(n)* returns the value of node *n*. Function *left(n)* returns the left node of node *n*. For example, the left node of node *t* is node *a* (see figure 3). Function *down(n)* returns the bottom node of node *n*. For example, the bottom node of node *d* is node *c* (see figure3). *hash* is a one-way collision-free hashing

function and || is the concatenation operation. Algorithm 2 applies to each node associated to the newly inserted element starting from the node at level 0 until the plateau node. Let us consider node *r* in figure 3 (index 24, value h_{24} and level 0) and node *e* (index 23, value h_{23} and level 0). The label of node *r* is equal to the hash of the value of node *r* (h_{24}) concatenated to the label of node *e* (line 5). Now, let us consider node *s* (index 24, value h_{24} and level 1) and node *d* (index 22, value h_{22} and level 1). The label of node *s* is equal to the hash of the label of node *r* concatenated to the label of node *d* (line 10). Let us consider also node *b* (index 21, value h_{21} and level 0) and node *p* (index 20, value h_{20} and level 0). Node *p* is not a plateau node, so the label of node *b* is equal to its value h_{21} (line 3). Finally, let us consider node *d* (index 22, value h_{22} and level 1) and node *c* (index 22, value h_{22} and level 0). The label of node *d* is equal to the label of node *c* (line 8).

Algorithm 3. Tail proof computation

```
{n is initialized to the plateau node of the element}
1:tp := {}
2:While right(n) != null
3:    n := right(n)
4:    If down(n) = null then
5:       append value(n) to TP
6:    Else
7:       append label(down(n)) to TP
8:    While top(n) != null :
9:       n: = top(n)
```

Algorithm 3 is used to compute the tail proof (*tp*) of elements which were inserted during the round. Function *right(n)* returns the right node of node *n*. For example, the right node of node *d* is node *s* (see figure 3). Function *top(n)* returns the node on top of node *n*. For example, the top node of node *s* is node *t* (see figure3). Computation of the tail proof of element *h* starts from the plateau node associated to element *h* (in algorithm 3, *n* is initialized to the plateau node of element *h*). Figure 3 shows that the tail proof of element h_{21} consists of h_{22} (that will be used during the verification to compute the label of node *c*), the label of node *r* (that will be used during the verification to compute the label of node *s*) and the label of node *x* (that will be used during the verification to compute the label of node *y*).

3.3 Verification Scheme

We call the *traversal chain* of element *h* the sequence of nodes whose labels have to be computed from *h* in order to determine the round token (label of the ending node *Et*). An example of such a chain is given by the thick nodes in Figure 3. They represent the traversal chain of element h_{21}. The final timestamp consists of the digest *h*, the index of *h*, the head proof of *h*, the tail proof of *h* and the round token. It contains all the necessary information to compute the labels of the nodes that are in the traversal chain of *h*. The verification process succeeds if the last label of the computed sequence of labels is equal to the round token. If not, the verification fails.

Algorithm 4. Verification

```
1: {h is the request, i_h the index of h}
2: label := h
3: index := i_h
4: level := 0
5: While TP != {}
6:    For i from level to height(index) :
7:       If hasPlateauOnLeft(index, i) then
8:          If leftIndex(index, i) < i_h then
9:             value := hash(label||getPrec())
10:         Else
11:            value := hash(getNext()||label)
12:   level := i.
13:   index := getNextIndex(index).
14:While HP != {} :
15:   label := hash(label||getPrec()).
16:If label = token then return TRUE
17:Else return FALSE
```

Algorithm 4 describes the verification process. Regarding that algorithm, we need to define the following functions:

- *height(index)*[4] that returns the level of the plateau node at position *index*
- *leftIndex(index, level)*[5] that returns the index of the left node of node of index *index* and level *level*
- *hasPlateauOnLeft(index, level)*[6] that indicates if the node of index *index* and level *level* has a plateau node on its left.
- *getNext()* that extracts the next label from the tail proof,
- *getPrec()* that extracts the next label from the head proof,
- *getNextIndex(index)*[7] that returns the index of the node whose label is the next label to be extracted by *getNext()*. That index can be computed from *index*.

In algorithm 4, h denotes the request and i_h the index of h (included in the timestamp). *token* denotes the round token included in the timestamp. Variable *label* denotes the label of the *current* node in the traversal chain. It is initialized to h. As we can see, the index of the request in the skip list is a parameter of algorithm 4. If the TSA would lie on the index of the request, then the verification would fail since it would not be possible to compute the labels of the nodes belonging to the traversal chain. Since the head proof is returned as a signed acknowledgement immediately after the request was received, the TSA cannot reorder elements in the skip list even before publishing the round token. Figure 3 shows the verification process for h_{21} (i_{h21} = 21). Labels of thick nodes are computed during the verification process. Variable *label* is initialized to h_{21}. Initial node of level 0 and index 21 is node b. Index of left

[4] Since we are dealing with perfect skip lists, the height h of any element can be computed from its index i : $i = 2^h * k$ where $HCF(2, k) = 1$.

[5] Since we are dealing with perfect skip lists, the left node of a node of index i and level l has an index $j = i - 2^l$.

[6] Consider node n of index i and level l. Consider k such that $i - 2^l = k * 2^l$. Since we are dealing with perfect skip lists, if $HCF(2,k) = 1$ then the left node of n is a plateau node.

[7] Since we are dealing with perfect skip lists, the next index j can be computed from the current index i: $j = i+2^h$, where h is the height of the element at position i.

node of node *b* is $21\text{-}2^0$ (=20). Left node of node *b* is not a plateau node. Therefore, label of node *b* is equal to the value h_{21} contained in variable *label*. Node *b* is a plateau node. Therefore, the next node processed by algorithm 4 is node *c* of index $21+2^0$ (=22) and of level 0. Index of left node of node *c* is $22\text{-}2^0$ (=21). Left node of node *c* is a plateau node. Therefore, label of node *c* is equal to the hash of the first label extracted from the tail proof (value of node *c*) concatenated to the label of node *b* ($hash(h_{22}||h_{21})$). Node *c* is not a plateau node. Therefore, the next node processed by algorithm 4 is node *d* of the same index 22 and of level 0+1 (=1). Left node of node *d* is not a plateau node. Therefore, label of node *d* is equal to label of node *c* ($hash(h_{22}||h_{21})$). Node *d* is a plateau node. Therefore, the next node processed by algorithm 4 is node *s* of index $22+2^1$ (=24) and of level 1. Index of left node of node *s* is $24\text{-}2^1$ (=22). Left node of node *s* is a plateau node. Therefore, label of node *s* is equal to the hash of the second label extracted from the tail proof (label of node *r*) concatenated to the label of node *d*. Node *s* is not a plateau node. Therefore, the next node processed by algorithm 4 is node *t* of index 24 and of level 2. Index of left node of node *t* is $24\text{-}2^2$ (=20). Left node of node *t* is a plateau node. Therefore, label of node *t* is equal to the hash of the label of node *s* concatenated to the first label extracted from the head proof (label of node *a*). Node *t* is not a plateau node. Therefore, the next node processed by algorithm 4 is node *u* of index 24 and of level 3. Left node of node *u* is not a plateau node. Therefore, label of node *u* is equal to label of node *t*. Node *u* is a plateau node. Therefore, the next node processed by algorithm 4 is the node of index $24+2^3$ (=32) and level 3. Note that in figure 3, there is no node of index 32. In fact, everything works as if 32 was the index of the ending element. Consequently, the next node is node *y*. Left node of node *y* is node *u* which is a plateau node. Index of left node is $32-2^3$ (=24). Therefore, the label of node *y* is equal to the hash of the third (and last) label extracted from the tail proof (label of node *x*) concatenated to the label of node *u*. Since node *y* is not a plateau node, the next node processed by algorithm 4 is the node of index 32 and level 4 i.e. node *z*. Index of left node of node *z* is $32\text{-}2^4$ (=16). Left node of node *z* is a plateau node. Therefore, label of node *z* is equal to the hash of the label of node *y* concatenated to the second label extracted from the head proof (label of node *o*). Since node *z* is not a plateau node, the next node processed by algorithm 4 is the node of index 32 and level 5 i.e. the node on top of node *z* i.e. the ending node. Index of left node of the ending node is $32\text{-}2^5$ (=0). Left node of the ending node is a plateau node. Therefore, label of the ending node is equal to the hash of the label of node *z* concatenated to the last label extracted from the head proof (label of node *q*). Since there is no more labels to extract, neither from the tail proof nor from the head proof, Algorithm 4 compares the last computed label to the round token included in the timestamp. If the two labels are equal then the verification succeeds. If not, the verification fails.

4 Performances

We have implemented a prototype of our time-stamping scheme (*http://chronos.univ-pau.fr*). We present the performances of our prototype. Table 1 compares the performances of our scheme and the other timestamping schemes. For each scheme, it shows the number of hashing operations which are necessary to process one request and the number of hashing operations which are necessary to process all the requests

belonging to one round (*n* is the number of processed requests). We can see that our scheme offers the same performances than the best timestamping schemes. In our scheme, each hashing operation comes after only one concatenation operation. BLS and TTS, need a greater number of concatenations before each hashing operation. Thus, time complexity in our scheme is much smaller than time complexity in those schemes. Furthermore, let us mention that the values presented for MTS, BLS and TTS stand for n being a power of 2. If it is not the case, then we need to add padding requests in MTS, BLS and TTS. This implies a significant number of useless concatenations and hashing operations.

Table 1. Timestamping costs

	Processing n requests	**Processing one request**
MTS[8]	*[n]c*[9] *- [n]h*[10]	*[1]c - [1]h*
LLS[11]	*[n]c - [n]h*	*[1]c - [1]h*
BLS[12]	*[3n/2]~c*[13] *- [n]h*	*[3/2]~c - [1]h*
TTS[14]	*[n (2+ lg(n)/2)-1]c - [2n-1]h*	*[lg(n)/2+2]~c-[2]h*
Chronos	*[n]c - [n]h*	*[1]c - [1]h*

We also focus on the size of timestamps and on the verification cost. For each scheme, table 2 shows the number of digests included in a timestamp (*n* is the total number of timestamps delivered in one round) and table 3 shows the number of concatenations and hashing operations necessary to verify a given timestamp. We can see that our scheme offers the same optimal performances than MTS. For example, for 10^7 delivered timestamps, each timestamp produced by Chronos includes 25 digests in the worst case. The verification requires 25 concatenations and 25 hashing operations. In BLS scheme, for the same number of requests, each timestamp includes about 100 digests, and verifying a timestamp requires about 100 concatenation operations and 67 hashing operations. In TTS scheme, timestamps contain 25 digests, and the verification needs about 37 concatenations and 25 hashing operations.

Table 2. Size of timestamps

	MTS	**LLS**	**BLS**	**TTS**	**Chronos**
Size of Timestamps	*lg(n) +1*	*n-1*	*4 lg(n)*	*lg(n)+1*	*lg(n)+1*

[8] MTS for Merkle Tree Scheme.
[9] *[i]c : i* is the exact number of concatenation operations.
[10] *[i]h : i* is the exact number of hashing operations.
[11] LLS for Linear Linking Scheme.
[12] BLS for Binary Linking Scheme.
[13] *[i]~c* : *i* is the average number of concatenation operations.
[14] TTS for Threaded Tree Scheme.

Table 3. Verification costs

	Verification
MTS	*[lg(n)+1]c - [lg(n)+1]h*
LLS	*[n-1]c - [n-1]h*
BLS	*[9/2 lg(n)-3]~c - [3lg(n)-2]~h*[15]
TTS	*[1+3/2 lg(n)]~c - [lg(n)+1]h*
Chronos	*[lg(n)+1]c - [lg(n)+1]h*

We could also compare our scheme to existing authenticated dictionary based on skip lists [5][12][13][15][16][17]. The reader can refer to [5] for such a comparison.

5 Conclusion

In this paper, we define a new totally ordered linking scheme based on skip lists. Our scheme offers better performances than existing totally ordered timestamping schemes. Moreover, it is easy to implement. Our scheme is for a single server TSA. The main drawback of single server TSAs is that they are vulnerable to denials of service. In [7][8], we suggest some directions to implement a multi-server timestamping system. The main idea used in [7][8] is to randomly choose k servers among n. In a future work, we plan to develop a distributed version of our scheme based on skip lists, which would use this concept of k among n.

References

[1] Ansper, A., Buldas, A., Willemson, J.: General linking schemes for digital time-stamping. Technical Report (1999)

[2] Bayer, D., Haber, S., Stornetta, W.: Improving the efficiency and reliability of digital time-stamping. In: Sequences 1991: Methods in Communication, Security and Computer Science, pp. 329–334 (1992)

[3] Benaloh, J., De Mare, M.: Efficient Broadcast time-stamping. Technical report 1, Clarkson University Department of Mathematics and Computer Science (1991)

[4] Benaloh, J.C., de Mare, M.: One-way accumulators: A decentralized alternative to digital signatures. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 274–285. Springer, Heidelberg (1994)

[5] Blibech, K., Gabillon, A.: A New Timestamping Scheme Based on Skip Lists. In: Proc. Of the 2006 International Conference on Computational Science and its Applications (Applied Cryptography and Information Security Workshop). Mai 2006, Glasgow, UK (2006)

[15] *[i]~h* : *i* is the average number of hashing operations.

[6] Blibech, K., Gabillon, A.: Authenticated dictionary based on skip lists for timestamping systems. In: Proc. of the 12th ACM Conference on Computer Security, Secure Web Services Workshop (2005)
[7] Bonnecaze, A., Liardet, P., Gabillon, A., Blibech, K.: A Distributed time stamping scheme. In: Proc. of the conference on Signal Image Technology and Internet based Systems (SITIS 2005), Cameroon (2005)
[8] Bonnecaze, A., Liardet, P., Gabillon, A., Blibech, K.: Secure Time-Stamping Schemes: A Distributed Point of View. Annals of Telecommunication 61(5-6) (2006)
[9] Buldas, A., Laud, P.: New Linking Schemes for Digital Time-Stamping. In: First International Conference on Information Security and Cryptology (1998)
[10] Buldas, A., Lipmaa, H., Schoenmakers, B.: Optimally efficient accountable time-stamping. In: Imai, H., Zheng, Y. (eds.) PKC 2000. LNCS, vol. 1751, pp. 293–305. Springer, Heidelberg (2000)
[11] Buldas, A., Laud, P., Lipmaa, H., Villemson, J.: Time-stamping with binary linking schemes. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 486–501. Springer, Heidelberg (1998)
[12] Goodrich, M., Tamassia, R., Schwerin, A.: Implementation of an authenticated dictionary with skip lists and commutative hashing (2001)
[13] Goodrich, M., Tamassia, R.: Efficient authenticated dictionaries with skip lists and commutative hashing. Technical report, J. Hopkins Information Security Institute (2000)
[14] Haber, S., Stornetta, W.S.: How to Time-stamp a Digital Document. Journal of Cryptology: the Journal of the International Association for Cryptologic Research 3(2) (1991)
[15] Maniatis, P., Baker, M.: Secure history preservation through timeline entanglement. Technical Report arXiv:cs.DC/0202005, Computer Science department, Stanford University, Stanford, CA, USA (2002)
[16] Maniatis, P., Giuli, T.J., Baker, M.: Enabling the long-term archival of signed documents through Time Stamping. Technical Report, Computer Science Department, Stanford University, California, USA (2001)
[17] Maniatis, P.: Historic Integrity in Distributed Systems. PhD thesis, Computer Science Department, Stanford University, Stanford, CA, USA (2003)
[18] Massias, H., Quisquater, J.J., Serret, X.: Timestamps: Main issues on their use and implementation. In: Proc. of IEEE 8th International workshop on enabling technologies: Infrastucture for collaborative enterprises (1999)
[19] Merkle, R.C.: Protocols for public key cryptosystems. In: IEEE Symposium on Security and Privacy (1980)
[20] Merkle, R.C.: A certified digital signature. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 218–238. Springer, Heidelberg (1990)
[21] Pugh, W.: Skip lists: a probabilistic alternative to balanced trees. Communications of the ACM, 668–676 (1990)

QFCP: A Router-Assisted Congestion Control Mechanism for Heterogeneous Networks (Extended)*

Jian Pu and Mounir Hamdi

Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Hong Kong, China
{pujian,hamdi}@cse.ust.hk

Abstract. Most existing end-to-end congestion control protocols employ packet loss or round-trip delay to imply network congestion. However, this kind of implicit signal mechanism may not work well in heterogeneous networks. Recently some router-assisted congestion control protocols are proposed to address this challenge and Quick Flow Control Protocol (QFCP) is one of them. QFCP allows flows to start with high initial sending rates and to converge to the fair-share rate quickly based on feedback from routers. The rate allocation algorithm is quite simple and only needs to be run periodically by routers. We have implemented QFCP in ns-2. Simulations have been done to address the issues such as flow completion time of Poisson-arriving Pareto-distributed-size flows, adaptability to changing flow numbers, fairness on flows with different RTTs, robustness to non-congestion packet losses, and performance on multiple bottleneck links. The preliminary results are promising.

1 Introduction

Congestion occurs when the packets injected into the network are more than that it is capable to deliver. In this situation, if the senders do not slow down their sending rate to relieve the traffic load, more and more packets will be dropped by the routers and little useful work can be done, which is called congestion collapse. Therefore, in order to make the network work stably and effectively, congestion control mechanisms have to be employed. Traditionally most of them are designed based on the end-to-end concept, only using the signals that can be measured at the end-systems to sense the possible network congestion. Some protocols use packet loss as the signal of network congestion, such as TCP, HighSpeed TCP, and Scalable TCP. Others employ increasing round-trip delay as the implication of congestion, such as TCP Vegas, and FAST TCP. However, the assumption that packet loss or delay increase indicates network congestion does not always hold when the path includes wireless links. Some factors other than congestion such as link bit-error rate, unstable channel characteristics, and user mobility may also contribute to packet loss or round-trip delay variety. For example, studies have shown that TCP performs poorly in wireless networks due

* This research was supported in part by the Hong Kong Research Grant Council under Grant RGC HKUST6260/04E.

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 115–125, 2009.

to its inability to distinguish packet losses caused by network congestion from those attributed to transmission errors. Much valuable bandwidth resource is wasted since TCP unnecessarily reduce its congestion window when non-congestion-related loss happens.

Recently some router-assisted congestion control protocols have been proposed to address this challenge, such as XCP [1], VCP [2], and RCP [3]. XCP is the first successfully designed protocol of this kind. It outperforms TCP in both conventional and high bandwidth-delay networks, achieving fair bandwidth allocation, high link utilization, small standing queue size, and low packet drops. VCP can be treated as a simpler version of XCP. It uses only 2 bits to encode the feedback from routers and is easier to deploy in the Internet. RCP proposes to use per-flow rate instead of per-packet window adjustment as the feedback from routers. The advantage of router-assisted congestion control mechanism is that routers are the place where congestion happens and only routers can give precise feedback on the network condition. The disadvantage is the deployment difficulty and the additional workload on routers. So there is a trade-off between the performance and the complexity. If router-assisted scheme does not outperform end-to-end scheme significantly, end-to-end scheme is preferred since it is easier to deploy in the Internet.

Another issue is that Additive Increase Multiplicative Decrease (AIMD) algorithm is too conservative for flows in high bandwidth-delay product networks. This becomes increasingly important as the Internet begins to include more high-bandwidth optical links and large-delay satellite links. Research on the Internet traffic has revealed an important feature at the flow level: most of the flows are very short, while a small number of long flows account for a large portion of the traffic [4], [5]. This is known as the heavy-tailed distribution. But TCP models flows as "long-lived flows", which means that all flows are assumed to be long enough to reach their fair-share sending rates before finishing. This assumption is acceptable when the Internet is mainly composed of low-speed links. However, if most flows are short as in today's high-speed networks, can the congestion control algorithm built in TCP continue to work efficiently? The answer is no. On the one hand, a short flow always starts with a very low initial sending rate and very likely finishes before reaching its fair-share rate. The duration of a short flow may be significantly prolonged due to packet loss, which causes timeout and packet retransmission [6]. On the other hand, a large number of short flows also adversely impact the performance of long flows. [7] shows that randomly generated sequence of short flows may reduce the throughput of long flows up to 10%, and some special pattern of short flows can even cause greater reduction (>85%). The reason is that short flows spend most of their lifetime in the Slow Start phase when their congestion windows increase exponentially. Thus, a burst of short flows can rapidly capture a great portion of the bandwidth and driven long flows into timeout and window-halving. But the AIMD algorithm grabs the available bandwidth very slowly and makes the time of converging to the fair-share rate very long for the long flows.

As the Internet keeps evolving into a high bandwidth-delay product (BDP) network, more and more flows are becoming short flows. And the adverse impact of TCP congestion control may become increasingly severe. We need to design a new congestion control protocol for the high-speed networks achieving the following metrics:

- For short flows, they can get high initial sending rates at the startup so that they can finish quickly ("quick start").
- For long flows, they can converge to the fair-share sending rate quickly and have maintainable high throughput ("quick convergence").

In most currently proposed congestion control protocols, each new flow starts with a low sending rate and then probes the network for the unused bandwidth. They only show that they can achieve fair rate allocation for long-lived flows. But short flows may suffer since they are not long enough to compete for fair bandwidth allocation.

If the router can find a way to compute the number of active flows and assign the fair-share rate to each flow, there will be no need to let flows wait for many RTTs to probe the network by themselves. Hence, we try to design a router-assisted congestion control protocol similar to XCP and RCP. The sending rate of each flow is controlled by routers along the path. The router senses the degree of congestion periodically and calculates a global fair rate for all flows passing through it. A new flow will start with the same sending rate as the other ongoing flows because they get the same feedback from routers. Additionally we want the routers:

- Do not store any per-flow information.
- Do not maintain any large hash tables or do any hash computation (vs. hash-based flow counting algorithm).
- Do not do complex per-packet calculation on routers.

2 Protocol Design

2.1 Framework

The basic idea is to establish a feedback control system in the network. Periodically a router checks the input traffic load and the queue size to sense the current network condition. Then it uses this information to generate feedback to control the sending rate of all flows through it.

We use a similar framework of Quick-Start [8], but we extend the rate-request-and-grant mechanism to the whole lifetime of a flow: (1) The sender includes a Rate Request field in the header of each outgoing packet and sets the initial value of this field to be the desired sending rate of this sender; (2) When the packet reaches a router, the router compares the value in Rate Request field with the router's own fair-share rate and puts the smaller one back into that field; (3) On receiving the packet, the receiver copies the Rate Request field into the Rate Feedback field of the corresponding ACK packet and sends it back to the sender; (4) When the sender receives the ACK packet, it reads the value in the Rate Feedback field and adjusts its sending rate accordingly.

The sender sets the size of its congestion window according to the rate feedback using the formula:

$$cwnd = rate * RTT\,, \tag{1}$$

where *cwnd* is the congestion window size, *rate* is the rate feedback, and *RTT* is the moving average round-trip time measured by the sender. This is because congestion window governs the flow throughput and its size (*cwnd*) determines the number of packets that can be outstanding within one RTT.

2.2 Rate Allocation Algorithm

The rate allocation algorithm is used by the router to compute the fair-share rate R periodically. One router maintains only one global fair-share rate for each output interface. This rate R is the maximum allowed rate for flows going through this interface during the current control period T. T is set to be the moving average of RTTs of all packets. The current number of flows through this interface is estimated using the aggregate input traffic rate and the fair-share rate assigned in the last control period. And the fair-share rate is updated based on the flow number estimation as follows.

$$N(t) = \frac{y(t)}{R(t-T)}, \tag{2}$$

$$R(t) = \frac{C - \beta \cdot \frac{q(t)}{T}}{N(t)}, \tag{3}$$

where $N(t)$ is the estimation of the flow number, $y(t)$ is the input traffic rate during the last control period, $R(t)$ is the fair-share rate, C is the bandwidth capacity of the output link, $q(t)$ is the queue size, β is a constant parameter, T is the control period.

This parameter β can be set as a policy by the router administer. A large value of β means one wants to drain up the queue more aggressively. The theoretical analysis of the impact of this parameter is left to the future work. Currently we set the value of β as 0.5 in our ns-2 implementation.

2.3 Technical Details

Burstiness Control: When designing congestion control protocols for large BDP networks, researchers often find that pure window-based rate control is not enough and additional burstiness control is needed (e.g., FAST TCP [9]). This is because window control is too rough and may trigger a large burst of packets injected into the network all at once (e.g., in the Slow Start phase of TCP). But such violent increase of congestion window is sometimes unavoidable in order to achieve good responsiveness and high throughput, especially in large BDP networks. In QFCP, we use the rate-based pacing to avoid possible burstiness caused by a sudden increase of congestion window. So although the sender may set a large window as approved by the rate feedback in the first ACK packet, it still needs to pace these packets out in one RTT based on the assigned sending rate. Thus, congestion window controls how many packets can be sent in one RTT, while burstiness control paces these packets out in a smooth way.

Rate Stabilization: If the assigned fair-share rate changes too quickly, the formula (2) we use to estimate the number of flows based on the previously assigned rate may fail. The reason is that there may be flows with RTT longer than the control period T using $R(t-2T)$ instead of $R(t-T)$ as the sending rate. So for the sake of the accuracy of flow number estimation, we don't want the rate assigned for two consecutive intervals to be very different. Thus, in order to stabilize the rate, we use the average of the current computed rate and the most recently assigned rate as the new rate.

Reaction to Packet Loss: If the queue of an interface is overflowed, the router will drop packets and add the number of dropped packet to $q(t)$ as $q(t)$ in formula (3), because this is the real queue size that should be drained during the next interval. The

router will use this "virtual queue size" in formula (3) to compute the new rate. The sender just retransmits the dropped packets without any further reaction. This is very different from the loss-based congestion control protocols, which will adjust the congestion window if encountering packet loss. So in QFCP, it is very easy to differentiate the two kinds of packet loss: one is due to the congestion; the other is due to transmission error. Because the rate is totally determined by the routers, the routers will adjust the rate according to the degree of congestion. There is no need for the end-systems to guess whether congestion happens or not when they encounter packet loss.

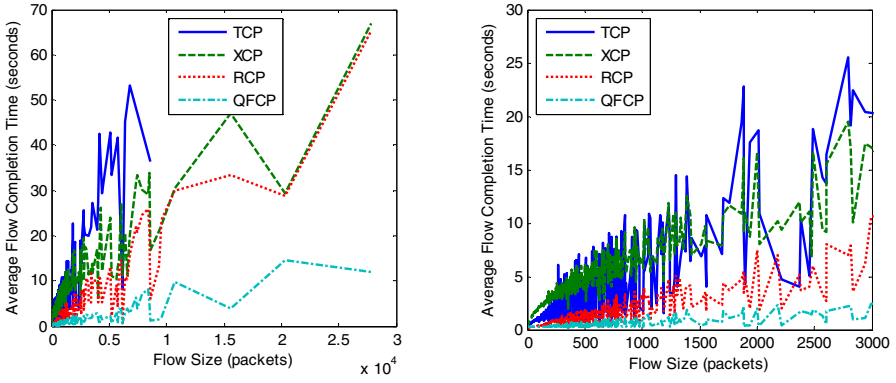
3 Simulation and Evaluation

In this section, we evaluate the performance of QFCP through extensive simulations using the packet-level network simulator ns-2 [10]. Unless specified otherwise, we use the dumb-bell network topology, where senders and receivers reside on each hand side and all flows go through the same bottleneck link. Simulations are run long enough to ensure the system has reached a consistent state. The parameter β of QFCP is set to 0.5. The buffer size on routers is set to be the delay-bandwidth product. The data packet size is 1000 bytes. And we use RED as the queue management scheme for TCP.

3.1 Flow Completion Time

For fixed-size flows (e.g., FTP, HTTP), the most attractive performance criterion is the flow completion time (FCT). Users always want to download files or web pages as fast as possible especially when they have paid for high-speed access links. Since a large number of flows in the Internet are short web-like flows, it is also important to investigate the impact of such dynamic flows on congestion control. Here we simulate a scenario where a large number of Pareto-distributed-size flows share a single bottleneck link of 150 Mbps. This is a typical mathematical model for the heavy-tail distributed Internet traffic which is composed of a large number of short flows. The total flow number is 60000. The common round-trip propagation delay is 100 ms. Flows arrive as a Poisson process with an average rate of 625 flows per second. Flow sizes are Pareto distributed with an average of 30 packets and a shape parameter of 1.2. Thus, the offered traffic load on the bottleneck link can be estimated as: 8 ∗ packet_size ∗ mean_flow_size ∗ flow_arrival_rate / bandwidth = 1. Such high traffic load is often the situation where differentiates the performance of congestion control protocols. We record the size and completion time for each flow in the simulation, then average the flow completion time for flows with the same size. Each simulation is conducted for each protocol: TCP-Reno, XCP, RCP, and QFCP. The scenario settings and the input data (i.e., the size and arriving time of each flow) are identical for each simulation. The results show that the Average Flow Completion Time (AFCT) in QFCP is significantly shorter than that in TCP, XCP or RCP.

For TCP, the AFCT is very oscillatory against the flow size. The reason is that although the exponential increase of congestion window in Slow Start does help some short flows finish quickly, the duration of other flows are prolonged due to packet



(a) all flows (b) flows with size < 3000 packets

Fig. 1. Average flow completion time (AFCT) vs. flow sizes for Poisson-arriving Pareto-distributed-size flows. (a) is the global picture for all flows. (b) is a close look at short flows.

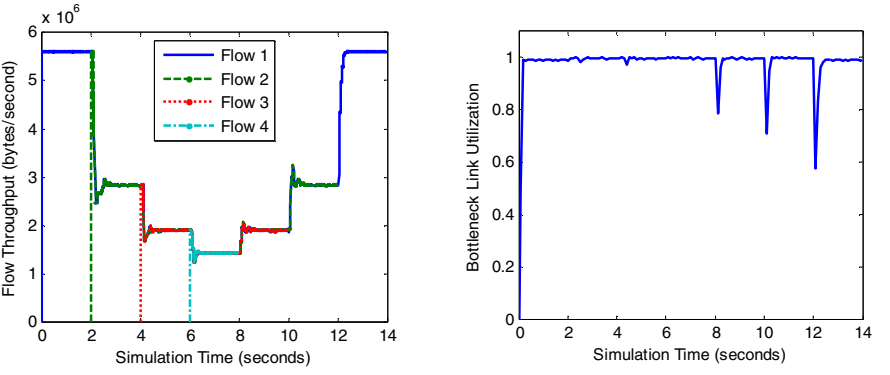
loss. And we should point out that Slow Start is not a good way to shorten the duration of flows, because it actually does not know the proper initial sending rate but just intends to fill up the buffer of routers and cause packet losses, which prolongs the duration of all flows.

For XCP, it does not use Slow Start. Instead, when new flows join, XCP tries to reclaim the bandwidth from the ongoing flows and reallocate it to the new flows little by little. For short flows, they may finish before reaching the fair sending rate. That is why the completion time of short flows in XCP is the longest. However, the AFCT against flow size in XCP is much more stable than in TCP because XCP flows experience fewer packet losses.

For RCP and QFCP, both of them give a high initial sending rate to new flows based on the feedback from routers and help short flows finish quickly. However, the formula used in RCP to estimate the number of flows holds only when the input traffic just fills up the link capacity C, otherwise it leads to wrong estimation of the flow number. This wrong estimation of flow number makes the rate allocation in RCP under-optimum and thus prolongs the FCT in general compared with QFCP.

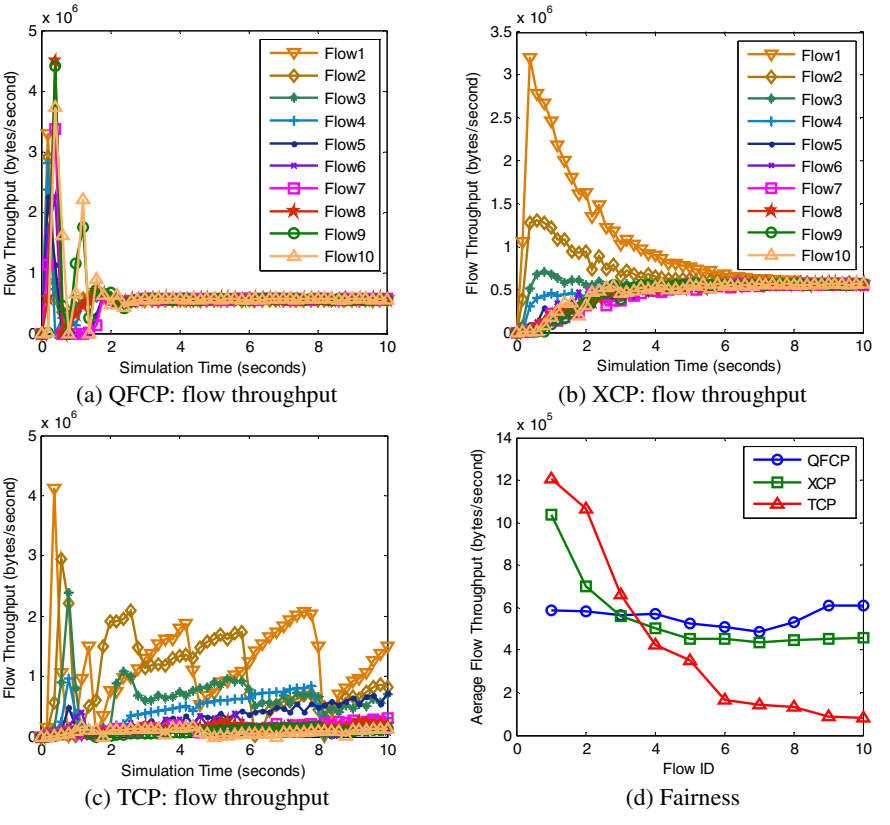
3.2 Adaptability to Changing Flow Numbers

For fixed-time flows or long-lived flows, we are more interested in the fairness of link bandwidth sharing among flows, especially when flow number changes causing bandwidth reallocation. In this simulation, four flows share a common bottleneck link of 45 Mbps. The common round-trip propagation delay of each flow is 40 ms. Flow 1-4 start at time 0, 2, 4, 6 seconds and stop at 14, 12, 10, 8 seconds respectively. The results show that QFCP can converge quickly to the fair-share sending rate as flows join and leave, and can maintain a high utilization of the bottleneck link. This simulation also shows a significant difference between QFCP and XCP: the high initial sending rate. In QFCP, any new flow starts with the same rate as the other ongoing flows, and then converges to the fair rate if this new flow is long enough. While in XCP, a new flow starts with a low sending rate and then converges to the fair rate.



(a) QFCP: flow throughput (b) QFCP: bottleneck link utilization

Fig. 2. Flow throughput and bottleneck link utilization



(a) QFCP: flow throughput (b) XCP: flow throughput

(c) TCP: flow throughput (d) Fairness

Fig. 3. Fairness on flows with different RTTs

That's why QFCP can help short flows finish much faster than XCP as demonstrated in the previous simulation.

3.3 Fairness on Variant RTTs

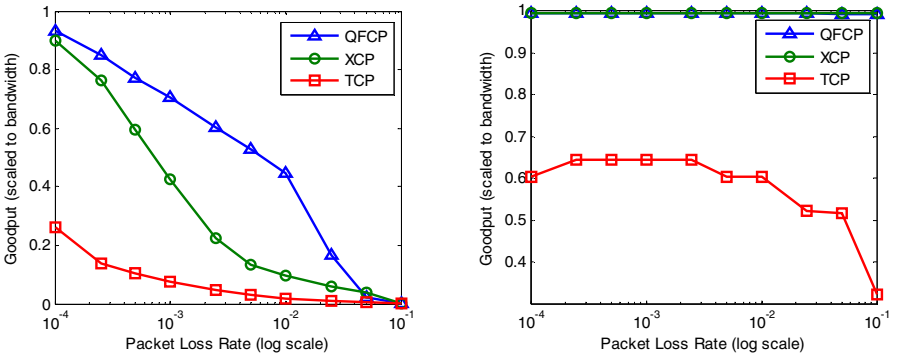
In this scenario we have 10 flows with different round-trip times sharing a common bottleneck link of 45 Mbps. The flow RTTs range from 40 ms to 220 ms in a step of 20 ms. All flows start at time 0 s and we wait until they converge to the steady state. The results in Fig. 3 show that although the RTTs of the flows are very different, QFCP fairly allocates the bottleneck bandwidth among the flows and converges to the fair-share rate even faster than XCP. For TCP, Fig. 3(c) (the legend is not shown for the sake of visibility) shows that some flows occupy most of the bandwidth while other flows keep sending at low rate. We then calculate the average throughput for each flow with different protocols. Fig. 3(d) shows that QFCP obtained the best fairness. It also confirms that TCP penalizes against flows with higher RTTs. Router-assisted protocols (QFCP, XCP) provide relatively fair bandwidth allocation for all flows although they use the average RTT as the control interval. However, for XCP, when the available bandwidth is large, flows with short RTTs increase their window much faster than flows with long RTTs, which causes transient unfairness as you can see in Fig. 3(b). This is because a flow with short RTT updates its window more frequently and gains more benefit from the positive feedbacks. While in QFCP, all flows get the same feedback on rate if they go through the same bottleneck no matter what RTTs they have. Thus, QFCP can converge to the fair-share rate faster than XCP.

3.4 Robustness on Lossy Link

This simulation tests for the sensitivity to non-congestion loss. The bottleneck link is 100 Mbps and can randomly generate packet losses at a fixed probability. The common RTT is 40 ms. Fig. 4(a) tests for data packet losses on the forward direction and Fig. 4(b) tests for ACK packet losses on the reverse direction. The loss rate ranges from 0.0001 to 0.1 and covers most typical loss ratios seen in a wireless network. Each simulation runs for 60 seconds and we record the highest acknowledged packet number seen at the sender to calculate the goodput. Goodput is the data packets that are successfully transferred (ACKed). It is necessary to differentiate goodput from throughput in this scenario since lossy link may cause massive packet retransmission but only goodput (highest ACKed packet number) is the work actually done from the user's vision.

The simulation results in Fig. 4 show that generally router-assisted congestion control performs better than TCP. ACK losses do not impact the goodput of XCP or QFCP very much because ACKs are cumulative and ACK loss can be recovered by subsequent ACKs. In contrast, data packet losses require retransmission. QFCP is more robust to data packet losses than XCP because the feedback in QFCP has more redundancy. For XCP, each ACK carries unique feedback on congestion window adjustment and the whole ACK packets in one RTT together give the correct value of aggregate feedback. Packet loss may cause difference between the actual congestion window size and the one expected by routers, especially when the lost ACK packet carries a large value of feedback. This may happen when the current window is small while the target window is large (e.g., the situation after a timeout). But for QFCP, since the feedback is directly the flow rate and this value is only updated once every control period, any ACK in the

current control period can give the correct window size to the sender. This kind of information redundancy gives high chance to prevent senders from starvation (keep sending at unnecessary low rate for a long time) in a lossy network.



(a) Data packet loss (b) ACK packet loss

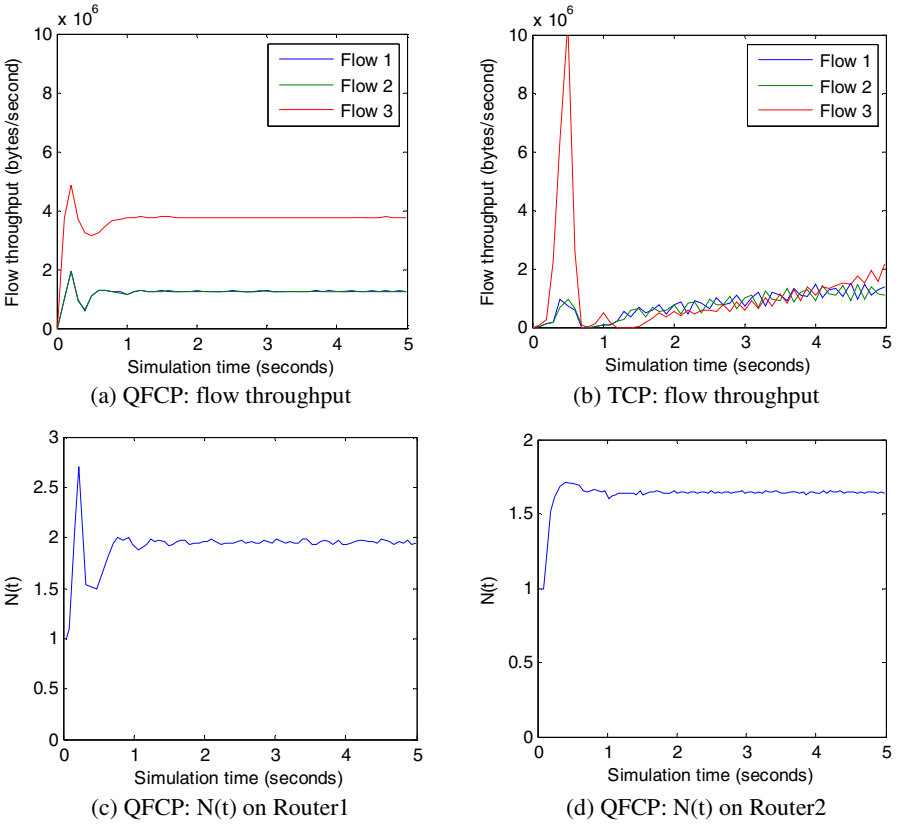
Fig. 4. Goodput on a lossy link

As mentioned before, packet loss is not treated as congestion signal in our scheme. For example, if 3 duplicate ACKs are received by the sender, TCP will halve the congestion window and retransmit the lost packet indicated by the dup-ACKs. This window shrinking may significantly reduce the flow rate and may be unnecessary if the packet loss is due to transmission media error instead of congestion. While in QFCP, upon 3 dup-ACKs, the sender will only retransmit the lost packet without shrinking the congestion window, unless it is told to do so explicitly by the feedback field of the ACK. This prevents unnecessary reduction on the flow rate. But for packet loss indicated by timeout event, QFCP still shrinks the congestion window as TCP does. This is a conservative procedure because no feedback is available at this point. But if the network is not congested (i.e., a non-congsetion loss), any subsequent ACK will restore the congestion window size to a proper value and the flow rate will be recovered immediately.

3.5 Flows Sharing Multiple Bottleneck Links

Previous simulations consider only one bottleneck link for simplicity. Now let's see whether this protocol can achieve max-min fairness when multiple bottleneck links are involved. In this simulation, there are two bottleneck links along the path. Flow1 and Flow2 share bottleneck Link1 of 20 Mbps. Flow3 is bottlenecked at Link2 of 50 Mbps. All of the three flows go through Link2. Thus, ideally the throughput of Flow1 and Flow2 should be 10 Mbps and the throughput of Flow3 should be 30 Mbps if congestion control works properly.

One may doubt that "Is one global rate for one output interface is enough? If some flows can not reach the sending rate assigned by a router, will the router still estimate the flow number correctly?" We should point out that the $N(t)$ estimated by a router is not the actual flow number but the equivalent flow number. And $N(t)$ can be any float number that greater than or equal to 1 (this lower limit 1 is set to avoid bandwidth



(a) QFCP: flow throughput (b) TCP: flow throughput

(c) QFCP: N(t) on Router1 (d) QFCP: N(t) on Router2

Fig. 5. Three flows sharing two different bottleneck links

oversubscription). Here is an example. For Router1, the estimated flow number is consistent with the real flow number 2, because both Flow1 and Flow2 can use up the rate assigned by Router1. However, for Router2, only Flow3 can send at the rate assigned by Router2, while the sending rate of the other two flows is limited somewhere else (Router1 in this case). Thus, the estimated flow number *N(t)* on Router2 is about 1.6, which is much less than the actual flow number 3. But this does not affect the stability and convergence of our protocol. Fig. 5(a) shows that all the three flows converge to their fair-share rate quickly and have maintainable high throughput. So we don't have such assumption that "all the flows must send at the assigned rate". The algorithm just wants to find out an equivalent flow number *N(t)* that best reflects the current situation of the network.

We also test the performance of TCP-Reno in this simple multiple-bottleneck-link scenario. The result shown in Fig. 5(b) confirms our previous analysis on TCP. The exponential increase of congestion window in the Slow Start phase quickly fills up the buffers of routers and causes packet drops. Then the throughput of all flows drop significantly due to packet retransmission and congestion window reduction. And flows quit the Slow Start phase and enter the congestion avoidance phase. But the AIMD algorithm grabs the available bandwidth very slowly restricting flows' sending rates.

4 Conclusions

Pure end-to-end congestion control scheme may unnecessarily reduce the sending rate when encountering non-congestion-related packet loss and underutilize the wireless networks. Router-assisted mechanism may help us design more robust congestion control protocols for better utilizing the heterogeneous networks. But we should notice that there is a trade-off between the performance and the complexity. In this paper, we introduce QFCP of this kind of protocol. It gives a high initial sending rate for any new flow as approved by routers along the path. The scheme can significantly shorten the completion time of both short flows and long flows. All flows can converge to the fair-share sending rate quickly whenever new flows join or old flows leave. And it also shows fairness for flows with different RTTs. It is easy for QFCP to differentiate non-congestion losses from congestion losses and thus prevent unnecessary window shrinking. The computation overhead on routers is acceptable and most calculations only need to do periodically. Future work may include implementing QFCP in Linux and deploying it in the real networks to test its performance. Establishing some mathematical models of QFCP and doing theoretical analysis on it are also desirable.

References

1. Katabi, D., Handley, M., Rohrs, C.: Congestion control for high bandwidth-delay product networks. In: Proceedings of ACM SIGCOMM 2002 the conference on applications, technologies, architectures, and protocols for computer communications, pp. 89–102. ACM Press, Pittsburgh (2002)
2. Xia, Y., Subramanian, L., Stoica, I., Kalyanaraman, S.: One more bit is enough. In: Proceedings of SIGCOMM 2005 the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Philadelphia, Pennsylvania, USA (2005)
3. Dukkipati, N., Kobayashi, M., Zhang-Shen, R., McKeown, N.: Processor sharing flows in the internet. In: de Meer, H., Bhatti, N. (eds.) IWQoS 2005. LNCS, vol. 3552, pp. 271–285. Springer, Heidelberg (2005)
4. Crovella, M.E., Bestavros, A.: Self-similarity in World Wide Web traffic: evidence and possible causes. IEEE/ACM Transactions on Networking 5, 835–846 (1997)
5. Claffy, K., Miller, G., Thompson, K.: The Nature of the Beast: Recent Traffic Measurements from an Internet Backbone. In: Proceedings of INET 1998 (1998)
6. Liang, G., Matta, I.: The war between mice and elephants. In: Proceedings of ICNP, pp. 180–188 (2001)
7. Ebrahimi-Taghizadeh, S., Helmy, A., Gupta, S.: TCP vs. TCP: a systematic study of adverse impact of short-lived TCP flows on long-lived TCP flows. In: Proceedings of INFOCOM 2005, vol. 2, pp. 926–937 (2005)
8. Jain, A., Floyd, S., Allman, M., Sarolahti, P.: Quick-Start for TCP and IP. IETF Internet-draft, work in progress (2005)
9. Jin, C., Wei, D.X., Low, S.H.: FAST TCP: motivation, architecture, algorithms, performance. In: Proceedings of INFOCOM 2004 the Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 4, pp. 2490–2501 (2004)
10. The network simulator ns-2, `http://www.isi.edu/nsnam/ns`

Formalization of Multimodal Languages in Pervasive Computing Paradigm

Arianna D'Ulizia and Fernando Ferri

IRPPS-CNR, via Nizza 128, 00198 Rome, Italy
{arianna.dulizia,fernando.ferri}@irpps.cnr.it

Abstract. The pervasive computing paradigm provides the user with a uniform computing space available everywhere, any time and in the most appropriate form and modality. These aspects produce the need for user interfaces that are usable, multimodal and personalized for each user. In this paper multimodality is discussed. In particular, features and computational issues of the multimodal interaction are analyzed in order to examine methodological aspects for the definition of multimodal interaction languages for pervasive applications. The multimodality is faced at a grammar level, rather than at a dialogue management level. This means that different unimodal inputs are considered as a unique multimodal input that is sent to the dialogue parser that uses the grammar specification to interpret it, rather than to be distinctly interpreted and then combined. The main objective of the paper is thus to explore multimodal interaction for pervasive applications through the use of a multimodal language rather than through the integration of several unimodal languages.

Keywords: Multimodal languages, relation grammars, pervasive computing systems.

1 Introduction

The increasing need to access virtual information and services everywhere and at any time has led pervasive computing to become the next generation of computing.

The pervasive computing paradigm regards almost every object in the everyday environment as a system that, equipped with embedded software and wireless communication facilities, is able to perceive, perform and control several tasks and functions, shifting computers into the background of user perception while exploiting them to support user activities and interactions. This paradigm focuses on people rather than machines, providing the user with a uniform computing space available everywhere, any time and in the most appropriate form and modality. In fact, users already rely on devices needing different modes of interaction, such as mobile phones, PDAs, tablet PCs and touch panels.

These features produce the need for user interfaces that are usable, multimodal and personalized to the greatest extent possible for each user. The term "multimodality" refers to the simultaneous or alternative use of several modalities and "modality" refers to the medium or channel of communication that conveys information [1].

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 126–136, 2009.

The use of multimodal user interfaces in a pervasive environment leads to three benefits over unimodality. First, multimodality improves accessibility to the device, as it provides users with the means to choose among available modalities according to the device's specific usability constraints. Second, multimodality improves accessibility by encompassing a broader spectrum of users, enabling those of different ages and skill levels as well as users with disabilities to access technological devices. Finally, it offers improved flexibility, usability and interaction efficiency.

Future user interfaces, through which users interact in a pervasive environment, must therefore enable both device and modality independence.

There is currently extensive activity in the field of multimodal user interfaces. An overview of a generic Multimodal Interaction Rendering System in a mobile environment is given in [2]. This system provides a framework for specifying, exchanging and rendering user interfaces and dialogs through a multimodal dialog and an interface specification language. Another approach to the specification and development of multimodal interfaces is ICARE [3]. This considers both pure modalities, described through elementary components, and combined modalities, specified by the designer through composition components.

Simon et al. [4] present a tool for the development of both graphical and multimodal web-based user interfaces for multiple devices.

In the field of multimodal interaction languages, activity is less intense, although [5] presents a study of the design of usable multimodal command or query languages.

All these works deal with multimodality at the dialogue management system level. This means that the different unimodal inputs are distinctly interpreted and then they are combined by the dialogue management system. In our work we aim at managing the multimodality at the grammar level. This means that the unimodal inputs are considered as a unique multimodal input that is sent to the dialogue parser that uses the grammar specification to interpret it.

There are several different parsing methodologies in the literature. These are applied to both textual and visual languages and can be extended to multimodal languages. Many of these methodologies employ a formal grammar, which may be considered a version of a more conventional grammar, modified to recognize syntactic patterns. Formal grammars such as Graph grammar [6][7], Relation-based grammar [8] and Attribute-based grammar [9][10] are able to capture language multi-dimensionality, which is one of the main characteristics of multimodal languages. As they are the result of the integration of different languages, each one may be inherently multi-dimensional, such as visual languages, that have a spatial arrangement of symbols instead of a sequential concatenation. Moreover, logical calculus and the use of logical approaches can improve the grammatical specification by representing aspects not directly specified in the grammar, exploiting hybrid approaches. In recent years, further purely logical approaches have been proposed to fully subsume the grammatical approach [11].

The focus of this paper is to analyze multimodal interaction features and characterize different modalities in order to examine methodological solutions for the definition of efficient, usable multimodal interaction languages and usable multimodal interfaces which allow the user to interact with technological systems in the context of pervasive applications. In particular, we make an analysis of existing grammars used for the specification of visual and textual languages and we study how to adapt these

grammars for the formalization of multimodal languages. In this way we aim at defining a single multimodal language obtained by integrating different unimodal languages. This area of research offers a large number of development possibilities and our main interest is the definition of a framework for the syntactic and semantic specification of multimodal languages.

The paper is organized as follows. Section 2 briefly describes features and issues of a multimodal interaction. Section 3 thoroughly describes the problems involved in the formal specification of multimodal interaction languages and the possibility of adapting existing grammars for visual and textual languages to this formal specification. Finally, conclusions and future work are given in section 4.

2 Features and Issues of Multimodal Interaction

Multimodal interaction [12] [13] provides the user with a way to interface with a system in both input and output. Figure 1 shows a simple schema that illustrates the basic architecture of a multimodal interaction system.

Characteristics of multimodal interaction relevant to computational modeling of user interfaces and interaction languages include:

Multiple modes: the modalities through which user and system can exchange information are manifold, and include speech, gesture, eye tracking, keyboarding, etc. An interaction modality can be defined [3] as a couple <d, L>, in which d is a physical device and L is the interaction language. Each modality provides a specific piece of information and taken together, they enable the command to be interpreted. Modalities can be classified as active or passive. The former is used when the user must explicitly perform an action with a device to specify a command. The latter is used when an explicit user action is not necessary to specify a command. The information specified by different modalities may be redundant.

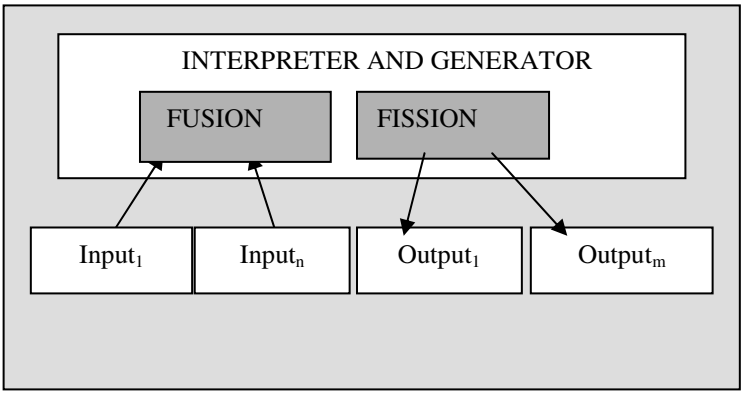


Fig. 1. Basic architecture of multimodal interaction system

Temporal constraints: in a multimodal dialogue there is no clear, definite instant in which the user finishes formulating the command.

In defining multimodal interaction languages, the input and output modes, temporal constraints and their related issues must be taken into account, as shown in Table 1.

To better understand the difficulties in formalizing languages for a multimodal environment, an explanation of these issues is given below.

Integrated interpretation of different inputs (fusion process): As a multimodal dialog involves the simultaneous use of multiple modalities, the user's input/commands must be interpreted through a fusion process. This integrates information from various input modalities by removing redundant or complementary information across the modalities and combining them into a complete command.

Synchronization of input modes: Timing is essential in conveying information during a multimodal interaction, so a tight synchrony among the various communicative modalities is required. This means that user inputs must be synchronized to deliver the correct information at the right time.

Table 1. Characteristics of a multimodal interaction and relative issues

	Characteristics		
	Multiple modes		Temporal constraints
	Input modes	Output modes	
Issues	integrated interpretation of different inputs (*fusion process*)	decomposition of different outputs (*fission process*)	gradual improvement in interpretation
	synchronization of input modes		*synchronization* of input modes

Decomposition of different outputs (fission process): The system has to find ways to integrate output through the various channels in order to provide the user with consistent feedback. This process is called fission, in contrast with multimodal fusion.

Gradual improvement in interpretation: The system must interpret the input while the interaction is ongoing and refine the interpretation when a new multimodal action is performed by the user.

Many works have focused on the development of a multimodal dialogue system which considers all the interaction features and issues described above. Gupta [14] outlines a method to collect input information supplied in different modalities, to determine when the user has finished providing input, to fuse the collected information to create a joint interpretation using an unification algorithm, and to send the joint interpretation to a dialogue manager that can perform reasoning. This method also considers temporal relationships between the modalities used during the interaction. Another comprehensive exploratory analysis of multimodal integration and synchronization patterns during pen-voice human-computer interaction is conducted by Oviatt et al. in [15].

In addition to these multimodality characteristics and issues, an environment to define multimodal languages should also satisfy the following features of pervasive systems:

***Adaptability**:* as a pervasive system has to support various situations (home, car, in-transit, field, privacy, noise, motion, postures, urgency, safety, etc.) and users (age, background, language, skills, disabilities), the environment should possess adaptation and customization features to facilitate its use by users with different activities, styles, skills and preferences.

***Portability**:* pervasive systems have to support various I/O hardware (handheld device, phone, paper, microphone, camera, data wall) and physical constraints (size, memory limitation, power consumption). The environment should therefore be adaptable to the capacities and constraints of every hardware platform supported.

3 Multimodal Interaction Languages and Their Formal Specification

The approach presented below allows the formalization of multimodal interaction languages. We are particularly interested in understanding how the described multimodal interaction issues and pervasive system features can be considered during the formalization phase.

3.1 Multimodal and Unimodal Languages

Let $M = \{ m_1, m_2, ..., m_n \}$ be the set of available modalities, according to the system's configuration. Each modality is associated with a unimodal interaction language l, that is a set of well-formed expressions (i.e. a conventional assembly of symbols) that convey meaning [3]. As previously described, $m_i = <d, l_i>$.

Let $L = \{ l_1, l_2, ..., l_n \}$ be the set of all unimodal interaction languages associated with all available modalities.

Consider the set **M'** of all possible subsets of the set M with cardinality greater than or equal to 2. Each subset represents a possible combination of available modalities.

M' = $\{\{m_1, m_2\},\{m_1, m_3\}, ..., \{m_1, m_n\},\{m_1, m_2, m_3\}, ..., \{m_1, m_2, m_n\}, \ldots, \{m_1, m_2, ..., m_n\}\}$

The set's cardinality is given by the formula:

$$2^n - (n+1) \tag{1}$$

Each element of **M'** is associated with one (or more) multimodal interaction language.

Let $L' = \{ l_1', l_2', ..., l_k' \}$ be the set of all multimodal interaction languages associated with all elements of **M'**.

Consider the example below, to clarify these definitions.

Let M = {speech, gesture, keyboarding, sketching} the set of available modalities. Each modality can be expressed through the couple <device, unimodal interaction language> in this way:

Speech=<microphone, natural language>
Gesture=<camera, signs language>
Keyboarding=<keyboard, keying in>
Sketching=<pen, drawing>

The set M' is composed of:

M' = {{speech, gesture}, {speech, keyboarding}, {speech, sketching}, {gesture, keyboarding}, {gesture, sketching}, {keyboarding, sketching}, {speech, gesture, keyboarding}, {speech, gesture, sketching}, {speech, keyboarding, sketching}, {gesture, keyboarding, sketching}, {speech, gesture, keyboarding, sketching}}.

To break the treatment down, we consider the element x ={m1, m2}={speech, sketching}. Although we refer to the integration of speech and sketching modalities, these considerations can be extended to any set of unimodal languages.

3.2 Formal Specification of Multimodal Languages

As previously described, unimodal interaction by speech involves natural language, while drawings are used to interact by sketching.

The composition of speech and sketches is based on placing discrete graphical objects such as lines, circles, arrows, icons, etc. on a drawing area alongside speech information about these graphical objects.

Our aim is to integrate these languages into a single multimodal language, to avoid the need for a combination of multiple unimodal interaction languages to achieve multimodal interaction. We therefore need to examine these two languages in more detail and introduce some definitions.

A *grapheme* is the atomic unit of a written language, such as letters, ideograms, or symbols.

A *phoneme* is the atomic unit of speech.

A *lexeme* is an atomic unit in a language that can designate a collection of graphemic and phonetic representations of words, phrases or graphic symbols.

Starting from graphemes or phonemes, the user can formulate a unimodal sentence to express a system command or a request through the sketching or speech modality. This sentence must be formulated in accordance with the production rules defined by the specific grammar, so that it can be recognized and understood by the system. The unimodal sentence thus represents an input message which is expressed in the specific unimodal interaction language.

An example of a unimodal sentence produced by the composition of phonemes is the phrase "John Smith", while an example of a unimodal sentence produced by sketching is shown in Figure 2.

Fig. 2. Example of unimodal sentence produced by sketching

Multimodal interaction enables the sequential, parallel or partially parallel/sequential formulation of multiple sentences belonging to different languages. It is essential to know the initial and final instants in which a sentence is produced, in order to measure its time-space in comparison with the other sentences and thus allow the synchronization and fusion of input sentences.

The syntax of the grammar for a language combining speech and sketching will operate on the properties, position, position relative to other objects and shape of graphical objects, but also on the words pronounced by the user and their temporal relationships. For this reason, it is more difficult to create a parser for a multimodal language than for a unimodal language. Several aspects must be considered:

- The symbols of each unimodal language cannot be represented by a linear ordering, as with sketching.
- Numerous relationships between symbols are possible, rather than a simple one-dimensional adjacency.
- Multimodal languages have a further difficulty due to synchronization of the different modalities.

The first aspect derives from the fact that some modalities, such as sketching, are not intrinsically one-dimensional. The second aspect, closely connected to the first, is that an infinite variety of relationships between symbols can be considered. In fact, each possible arrangement of two objects in a two-dimensional space can be considered as a different relationship between two objects. Finally, a sentence belongs to different unimodal languages and each of these components occurs sequentially, in parallel or partially in parallel/sequentially with respect to one another. In consequence, the sentence's meaning may differ according to the synchronization.

For instance, consider the speech and sketch sentences introduced above in the context of an electronic phone book application. In Figure 3.a, the user asks the system for John Smith's phone number by saying the words "John Smith". He/she then decides to phone John Smith and transmits this intention to the system by sketching

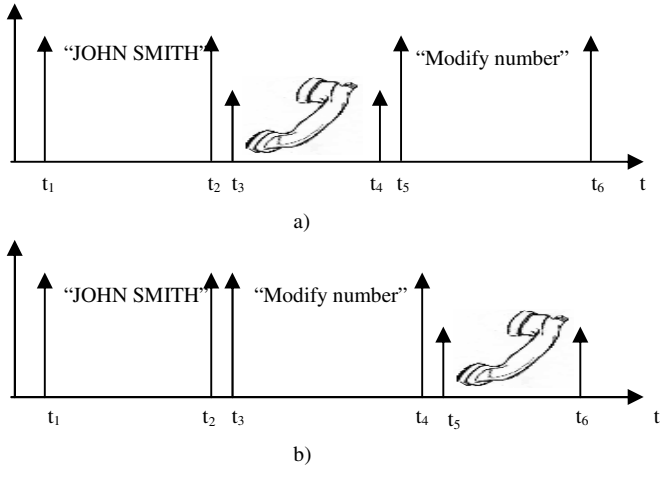


Fig. 3. Formulation of different sentences

the telephone icon. Finally, he/she decides to modify John Smith's telephone number by saying the words "Modify number". In Figure 3.b the user says the words "John Smith" "Modify number", and is then enabled to type in John Smith's new number. Finally, he/she sketches the telephone icon to inform the system of his/her intention to call John Smith.

These three aspects highlight the need for a grammar that is able to consider both spatial and temporal relationships among atomic units of the multimodal language and express their multi-dimensional aspects. To achieve this, we excluded the context-free grammar model due to its limited generative power (it is unable to generate graph languages). As multimodal languages require the integration and handling of different multimodal inputs with an often complex structure, a more powerful generative grammar model must be chosen for their formalization.

Multi-dimensional grammars are well suited to cover the formal definition of multimodal language syntax, as they yield a formalism that can encode the structure of a variety of multimodal inputs from different devices. This kind of grammar also uses attributes as an essential part of the parsing algorithm, as their values are crucial for syntactic analysis. For the purpose of this paper it is not necessary to analyse all different kinds of multi-dimensional grammars that has been presented in the literature, but we have selected two main classes that we hold as the most suitable for the formalization of multimodal languages: relation grammars [16] and attribute grammars [17]. In the next section we show an application of relation grammars to formalize multimodal languages.

The main shortcoming of grammar-based methods is that they provide little support in processing semantics.

To override this drawback, the use of a logical approach can improve grammatical specification by representing aspects not directly specified in the grammar. This hybrid approach, that provides a logical level over multidimensional grammars, is the most suitable for dealing with syntax without omitting the language's semantic aspects.

Before introducing a basic idea on how to formally specify a multimodal interaction language, it should be noted that a unimodal interaction language l can be formally described through a grammar as defined below.

Definition 1. A formal grammar is a quadruple, G = (I,T,X,P), where:

- I is a finite set of non-terminal symbols, also called variables alphabet;
- T is a finite set of terminal symbols not in I, also called object alphabet;
- X is a finite set of production rules;
- P is a start symbol in I.

Definition 2. The language of a formal grammar G = (I,T,X,P), denoted as L(G), is defined as all those strings over T, called sentences, that can be generated by beginning with the start symbol P and then applying the production rules in X.

Given any combination of modalities, e.g. an element **x** of the set **M'**, we want to create a formal grammar G'= (I',T',X',P') to define the multimodal language l' associated with the modalities belonging to **x.** Without loss of generality, we suppose that $x = \{m_1, m_2\} = \{<d_1, l_1>, <d_2, l_2>\}$,where l_1 is defined by the grammar $G_1 = (I_1,T_1,X_1,P_1)$ and l_2 is defined by the grammar $G_2 = (I_2,T_2,X_2,P_2)$.

Starting from Definition 1, the first element to be defined is the object alphabet T'. This set is composed of the union of object alphabets of l_1 and l_2, that is $T' = T_1 \cup T_2$.

The variables alphabet I' and set of production rules X' can be defined in relation to the specific multimodal inputs m_1 and m_2 and the adopted grammar. In any case, I' and X' also depend on the corresponding sets of the unimodal languages: I' depends on I_1 and I_2 while X' depends on X_1 and X_2.

In the next section we define better the elements of a formal grammar for a multimodal language l' by applying a specific kind of multidimensional grammars, i.e. the relation grammars.

3.3 Formal Specification of Multimodal Languages through Relation Grammars

The relation-based approach models a multimodal sentence, like that shown in Figure 4, as a set of lexemes T (terminal symbol) and a set of primitive relations R among these lexemes. The formal definition of multimodal sentence is given below.

Definition 3. Given an alphabet of terminal symbol T and a set of relations R, a multimodal sentence *m* over T and R is a pair $(A_T, R(A_T))$ such that A_T is a multiset of lexemes in T and $R(A_T)$ is a set of relation instances in R over lexemes in A_T.

As a running example consider the multimodal language that uses the speech and sketch modalities, such as the one shown in Figure 4. In the example we consider just the four terminal symbols introduced in the multimodal sentence of Figure 4 and a set of temporal relationships. Obviously the approach can be extended to consider a larger set of terminal symbols and a larger set of relationship including spatial and reference relationships. Therefore, to represent the multimodal sentence of Figure 4 we can use the following sets of lexemes and relations:

T = {"John", "Modify", "Number",  }

R = {*start_together, end_together, start_after, end_after, start_before, end_before, start_during, end_during, correct*}

and the multimodal sentence $m = (A_T, R(A_T))$ is represented in this way:

A_T = T and

$R(A_T)$ = (*start_before*(John, ), *end_during*(John, ), *start_before*(John, Number, *end_before*(John,Number), *start_before*(John,Modify), *end_before*(John, Modify), *start_before*( , Number), *end_during*( , Number), *start_before* ( ,Modify), *end_before*( ,Modify), *start_before*(Number,Modify), *end_before*(Number,Modify).

In this representation *start_before, end_during*, etc.. are relations and *end_before*(John,Number), *end_before*(John, Modify), etc.. are relation instances.

In order to characterize the multimodal sentences that have to be recognized by the parser we can introduce the definition of relation-based grammar.

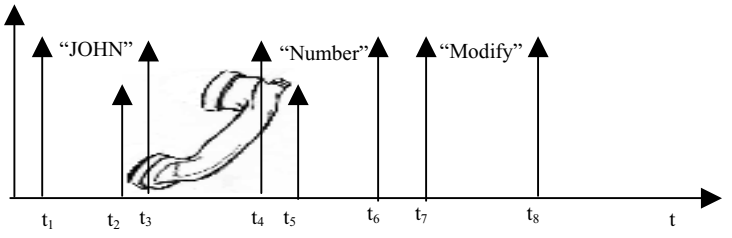


Fig. 4. An example of multimodal sentence

Definition 4. A relation grammar is a 6-tuple $G_r = (I,T,R,P,X,C)$ where:

- I is the set of non-terminal symbols,
- T is the set of terminal symbols,
- R is the set of relations,
- P is the start symbol in I,
- X is the set of production rules,
- C is the set of evaluation rules used to validate constraints.

Proceeding with the previous example, the grammar G_r that accepts multimodal sentences, like that shown in Figure 4, is the following:

$G_r = (I,T,R,P,X,C)$

where I = {P}, R and T are as above-mentioned, X contains the production rules:

P ::= {"John"}
P ::= {"John",$\mathcal{J}$}{correct("John",$\mathcal{J}$)}

and C contains the following rules:

correct("John",$\mathcal{J}$) :- {[start_before("John",$\mathcal{J}$) $\vee$ start_together("John"$\mathcal{J}$)] $\wedge$ [end_during("John",$\mathcal{J}$) $\vee$ end_together("John",$\mathcal{J}$) $\vee$ end_after("John",$\mathcal{J}$]}.

The correct relation, defined by the evaluation rule in C, expresses the fact that the starting lexeme "John" can be followed by the lexeme $\mathcal{J}$ and it can end during, together or after this one.

The relation-based approach has the advantage that it allows reasonably efficient parsing in the context-free case. However, it has the disadvantage that the relations are purely symbolic and are not interpreted arithmetically.

4 Conclusion

This paper discussed the formalization of multimodal languages. Considering the importance of multimodality to improve user interfaces and their usability, our main objective was to define a formal grammar to formalize multimodal languages. The paper considered some features and characteristics of multimodal interfaces and then discussed the use of both multidimensional grammars and logical approaches in the specification of multimodal languages. Future work would involve the development

of a complete framework to specify multimodal languages, investigating the possibilities of graph grammars, logical approaches or hybrid approaches.

References

1. Coutaz, J., Caelen, J.: A Taxonomy For Multimedia and Multimodal User Interfaces. In: Proceedings of the 1st ERCIM Workshop on Multimedia HCI, Lisbon (November 1991)
2. Mueller, W., Schaefer, R., Bleul, S.: Interactive Multimodal User Interfaces for Mobile Devices. In: 37th Hawaii International Conference on System Sciences (HICSS 2004) (2004)
3. Bouchet, J., Nigay, L.: Balzagette, D., ICARE: Approche à composants pour l'interaction multimodale. In: Actes des Premières Journées Francophones: Mobilité et Ubiquité 2004, Nice Sophia-Antipolis, France, pp. 36–43 (June 2004)
4. Simon, R., Wegscheider, F., Tolar, K.: Tool-Supported Single Authoring for Device Independence and Multimodality. In: 7th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI 2005) (2005)
5. Carbonell, N.: Towards the design of usable multimodal interaction languages. Journal on Universal Access in the Information Society (UAIS 2003) 2(2), 143–159 (2003)
6. G"ottler, H.: Graph grammars, a new paradigm for implementing visual languages. In: Eurographics 1989, pp. 505–516. ElsevierScience publishers, North-Holland (1986)
7. Göttler, H.: Graph grammars and diagram editing. In: Ehrig, H., Nagl, M., Rosenfeld, A., Rozenberg, G. (eds.) Graph Grammars 1986. LNCS, vol. 291, pp. 211–231. Springer, Heidelberg (1987)
8. Crimi, C., Guercio, A., Nota, G., Pacini, G., Tortora, G., Tucci, M.: Relation grammars for modelling multi-dimensional structures. In: IEEE Symposium on Visual Languages, pp. 168–173. IEEE Computer Society Press, Los Alamitos (1990)
9. Helm, R., Marriott, K., Odersky, M.: Building visual language parsers. In: ACM Conf. Human Factors in Computing, pp. 118–125 (1991)
10. Marriott, K.: Constraint multiset grammars. In: IEEE Symposium on Visual Languages, pp. 118–125. IEEE Computer Society Press, Los Alamitos (1994)
11. Marriott, K., Meyer, B., Wittenburg, K.: A survey of visual language specification and recognition. In: Marriott, K., Meyer, B. (eds.) Visual Language Theory, pp. 5–85. Springer, New York (1998)
12. Stivers, T., Sidnell, J.: Introduction: Multimodal interaction. Semiotica 156(1/4), 1–20 (2005)
13. Thrisson, K.R.: Computational Characteristics of Multimodal Dialogue. In: AAAI Fall Symposium on Embodied Language and Action, Massachusetts Institute of Technology, Cambridge, Massachusetts, November 10-12, pp. 102–108 (1995)
14. Gupta, A.: An adaptive approach to collecting multimodal input. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (2003)
15. Oviatt, S., DeAngeli, A., Kuhn, K.: Integration and Synchronization of Input Modes during Multimodal Human-Computer Interaction. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 415–422 (1997)
16. Crimi, C., Guercio, A., Nota, G., Pacini, G., Tortora, G., Tucci, M.: Relation grammars for modelling multi-dimensional structures. In: IEEE Symposium on Visual Languages, pp. 168–173. IEEE Computer Society Press, Los Alamitos (1990)
17. Helm, R., Marriott, K., Odersky, M.: Building visual language parsers. In: ACM Conf. Human Factors in Computing, pp. 118–125 (1991)

Generation of Spatial Decision Alternatives Based on a Planar Subdivision of the Study Area

Salem Chakhar and Vincent Mousseau

LAMSADE, University of Paris Dauphine,
Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France
{chakhar,mousseau}@lamsade.dauphine.fr
http://www.lamsade.dauphine.fr

Abstract. Outranking methods, a family of multicriteria analysis tools, cope better with the ordinal aspects of spatial decision problems. However, it is recognized that these methods are subject to computational limitations with respect to the number of alternatives. This paper proposes an approach to generate these alternatives based on a planar subdivision of the study area. The planar subdivision is obtained by combining a set of criteria maps. The result is a set of non-overlapping polygons/spatial units. Punctual, linear and areal decision alternatives, conventionally used in spatial multicriteria analysis, are then constructed as an individual, a collection of linearly adjacent or a collection of contiguous spatial units. This permits to reduce substantially the number of alternatives enabling the use of outranking methods.

Keywords: Planar subdivision, GIS, Multicriteria Analysis, Decision alternatives.

1 Introduction

Multicriteria analysis (MCA) tools are often used in spatial contexts to evaluate and compare a set of potential decision alternatives—often modeled through punctual, linear or areal entities and evaluated on several criteria—in order to select a restricted subset for implementation. They have been incorporated into geographical information systems (GIS) to enhance its modeling and analysis capabilities. The author in [9] provides a recent review on GIS-MCA integration covering the period 1990-2004. MCA methods are commonly categorized, based on the set of alternatives, into discrete and continuous. There are two families of methods within the discrete category: utility function-based approach and outranking-based approach.

Most of MCA based-GIS systems use utility function-based methods (e.g. [1]). These methods still dominate today and only few works (e.g. [10]) use outranking-based ones. Outranking methods cope better with spatial decision problems since they: (i) permit to consider qualitative evaluation criteria (in addition to quantitative ones) for which preference intervals ratios have no sense; (ii) permit to consider evaluation criteria with heterogenous scales that coding them into one

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 137–148, 2009.

common scale is very difficult or artificial; (iii) avoid the compensation between evaluation criteria; and (iv) require fewer amount of information from the decision maker (DM). But the major drawback of outranking methods (except those devoted to multicriteria sorting problems) is that they are not suitable for problems implying a large or infinite number of alternatives. Indeed, it is recognized that these methods are subject to computational limitations with respect to the number of alternatives [11] as most methods require pairwise comparison across all alternatives.

In this paper, we propose an approach to generate spatial alternatives based on a planar subdivision, that we call *decision map*, of the study area. The planar subdivision is obtained by combining a set of criteria maps. The result is a set of non-overlapping polygonal spatial units. Punctual, linear or areal decision alternatives are then constructed as an individual, a collection of linearly adjacent, or a collection of contiguous spatial units. This permits to reduce substantially the number of alternatives enabling the use of outranking methods.

The rest of the paper is as follows. Section 2 provides the background. Section 3 briefly introduces multicriteria analysis. Section 4 introduces the concept of decision map. Section 5 proposes solutions for the generation of spatial decision alternatives. Section 6 shows how the proposed solutions can be exploited to generate composed decision alternatives. Section 7 briefly presents some implementation issues. Section 8 concludes the paper.

2 Background

We consider only simple area, line and point features of $\mathbf{R}^2$. In the rest of the paper, the letters P, L, and Q are used to indicate point, line, and area features, defined as follows: (i) An area feature Q is a two-dimensional open points-set of $\mathbf{R}^2$ with simply connected interior Q° (with no hole) and simply connected boundary ∂Q; (ii) A line feature L is a closed connected one-dimensional points-set in $\mathbf{R}^2$ with no self-intersections and with two end-points. The boundary ∂L of L is a set containing its two endpoints and its interior L° is the set of the other points; and (iii) A point feature P is zero-dimensional set consisting of only one element of $\mathbf{R}^2$. The interior P° of a point feature P is the point itself and its boundary is empty (i.e. $\partial P = \emptyset$). Below, the symbol γ may represent anyone of the three feature types.

There are several proposals for classifying topological spatial relationships (see [5] for a comparative study of some classification methods). These classifications are based on the intersection of boundaries, interiors and exteriors of features. In [4] the authors introduce the CBM (Calculus-Based Method) based on object calculus that takes into account the *dimension* of the intersections. The authors provide formal definitions of five (`touch`, `in`, `cross`, `overlap`, and `disjoint`) relationships and for boundary operators. They also proved that these operators are mutually exclusive, and they constitue a full converging of all topological situations. In the following, we recall the definitions of the CBM.

Definition 1. *The* `touch` *relationship applies to all groups except point/point one:*

$$(\gamma_1, \texttt{touch}, \gamma_2) \Leftrightarrow (\gamma_1^\circ \cap \gamma_2^\circ = \emptyset) \wedge (\gamma_1 \cap \gamma_2 \neq \emptyset)$$

Definition 2. *The* `in` *relationship applies to every group:*

$$(\gamma_1, \texttt{in}, \gamma_2) \Leftrightarrow (\gamma_1 \cap \gamma_2 = \gamma_1) \wedge (\gamma_1^\circ \cap \gamma_2^\circ \neq \emptyset)$$

Definition 3. *The* `cross` *relationship applies to line/line and line/area groups:*

$$(L_1, \texttt{cross}, L_2) \Leftrightarrow (L_1 \cap L_2 \neq \emptyset) \wedge (dim(L_1 \cap L_2) = 0)$$

$$(L, \texttt{cross}, Q) \Leftrightarrow (L \cap Q \neq \emptyset) \wedge (L \cap Q \neq L)$$

Definition 4. *The* `overlap` *relationship applies to area/area and line/line groups:*

$$(\gamma_1, \texttt{overlap}, \gamma_2) \Leftrightarrow (dim(\gamma_1^\circ) = dim(\gamma_2^\circ) = dim(\gamma_1^\circ \cap \gamma_2^\circ))$$

$$\wedge(\gamma_1 \cap \gamma_2 \neq \gamma_1) \wedge (\gamma_1 \cap \gamma_2 \neq \gamma_2)$$

Definition 5. *The* `disjoint` *relationship applies to every group:*

$$(\gamma_1, \texttt{disjoint}, \gamma_2) \Leftrightarrow (\gamma_1 \cap \gamma_2 = \emptyset))$$

In order to enhance the use of the above relationships, the authors in [4] have defined operators able to extract boundaries from area and lines features. The boundary operator `b` for an area feature Q returns the circular line of ∂Q. The boundary operators `f` and `t` for a line feature return the two end-points features of L.

3 Multicriteria Analysis

In MCA the DM has to choose among several possibilities, called *alternatives*, on the basis of a set of, often conflicting, evaluation *criteria*. The set of alternatives A may be finite (or denumerable) or infinite. The MCA methods are categorized on basis of set A into *discrete* and *continuous*. In this paper we are concerned with the first category. Let $A = \{x_1, x_2, \cdots, x_n\}$ denotes a set of n alternatives. The evaluation criteria are factors on which alternatives are evaluated and compared. Formally, a criterion is a function g_j, defined on A, taking its values in an ordered set, and representing the DM's preferences according to some points of view. The evaluation of an alternative x according to criterion g_j is written $g_j(x)$. Let $F = \{1, 2, \cdots, m\}$ denotes the set of criteria indices.

To compare alternatives in A, we need to aggregate the partial evaluations (i.e. in respect to each criterion) into a global one by using a given *aggregation function*. Within discrete family, there are usually two aggregation approaches: (i) *utility function-based approach*, and (ii) *outranking relation-based approach*. In the rest of this paper we focalize on the second approach. Outranking methods

use partial aggregation functions. Indeed, criteria are aggregated into partial binary relation S, such that aSb means that "a is at least as good as b". The binary relation S is called *outranking relation*. The most known method in this family is ELECTRE (see [6]). To construct the outranking relation S, we compute for each pair of alternatives (x, y), a concordance indices $C(x, y) \in [0, 1]$ measuring the power of criteria that are in favor of the assertion xSy and a discordance indices $ND(x, y) \in [0, 1]$ measuring the power of criteria that oppose to xSy. Then, the relation S is defined as follows:

$$\begin{cases} C(x, y) \geq \hat{c} \\ ND(x, y) \leq \hat{d} \end{cases}$$

where $\hat{c}$ and $\hat{d}$ and a concordance and a discordance threshold, respectively. Often an exploitation phase is needed to "extract", form S, information on how alternatives compare to each other. At this phase, the concordance and discordance indices ($C(x, y)$ and $ND(x, y)$) are used to construct indices $\sigma(x, y) \in [0, 1]$ representing the credibility of the proposition xSy, $\forall(x, y) \in A \times A$. The proposition xSy holds if $\sigma(x, y)$ is greater or equal to a given *cutting level*, λ.

In spatial contexts, alternatives are often modeled through one of three spatial entities, namely *point* (P), *line* (L), or *area* (or *polygon*) (Q). For instance, in a facility location problem potential alternatives take the form of points representing different possible sites. This way of modeling generates a rich set of spatial alternatives. As consequence, outranking-based methods quickly reach their computational limits. Evaluation criteria are associated with geographical entities and relationships between entities and therefore can be represented in the form of maps. A *criterion map* is composed of a collection of spatial units; each of which is characterized with one value relative to the concept modeled. Mathematically, a criterion map $\mathbf{c_j}$ is the set $\{(s, g_j(s)) : s \in S_j\}$ where S_j is a set of spatial units and g_j is a mono-valued criterion function defined as follows:

$$\begin{aligned} g_j : S_j &\rightarrow E \\ s &\rightarrow g_j(s) \end{aligned}$$

E is an ordinal (or cardinal scale). One should distinguish a simple map layer from a criterion map. In fact, a criterion map models the preferences of the DM concerning a particular concept, which is often with no real existence while a simple map layer is a representation of some spatial real data. In practice, criteria are often of different types and may be evaluated according to different scales. Here we suppose that criteria are evaluated on the same ordinal scale.

4 Concept of Decision Map

4.1 Definition

A *decision map* is a planar subdivision represented as a set of non-overlapping polygonal spatial units that are assigned using a multicriteria sorting model, Γ_ω, into an ordered set of categories representing evaluation levels. More formally, a decision map $\mathbf{M}$ is defined as $\mathbf{M} = \{(u, \Gamma_\omega(u)) : u \in U, \omega \in \Omega\}$, where U is a set of homogenous spatial units and Γ_ω is defined as follows:

$$\begin{aligned} \Gamma_\omega : U &\rightarrow E \\ u &\rightarrow \Gamma_\omega[g_1(u), \cdots, g_m(u), w] \end{aligned}$$

where: (i) $E : [e_1, e_2, \cdots, e_k]$: (with $e_i \succ e_j$, $\forall i > j$) is an ordinal scale defined such that e_i, for $i = 1..k$, represents the evaluation of category C_i; (ii) $g_j(u)$: is the performance of spatial unit u in respect to criterion g_j associated with criteria map $\mathbf{c}_j$; (iii) Ω: is the space of possible values for preference parameters vector $\tau = (\tau_1, \tau_2, \cdots, \tau_v)$ associated with Γ_ω; and (iv) $\omega \in \Omega$: a single vector of preference parameters values.

Spatial units need to be non-overlapping and together constitue $\mathbf{M}$. Let $I = \{1, 2, \cdots, n\}$ be the set of the indices of the spatial units composing $\mathbf{M}$. Then, two conditions need to be verified:

$$\mathbf{M} = \bigcup_{i \in I} u_i, \quad \text{and} \quad u_i^\circ \cap u_j^\circ = \emptyset, \forall i, j \in I \wedge i \neq j.$$

The first condition ensures that the partition is total. The second one ensures that spatial units are non-overlapping. In addition, we generally suppose that the evaluations of connected spatial units are distinct, that is:

$$\partial u_i \cap \partial u_j \neq \emptyset \Leftrightarrow \Gamma_w(u_i) \neq \Gamma_w(u_j), \forall i, j \in I \quad \text{and} \quad i \neq j.$$

4.2 Construction of the Planar Subdivision

The construction of a decision map needs the superposition of a set of criteria maps. The result is an intermediate map $\mathbf{I}$ composed of a new set of spatial units that result from the intersection of the boundaries of the features in criteria maps. Each spatial unit u is characterized with a vector $\mathbf{g}(u)$ of m evaluations:

$$\begin{aligned} \mathbf{I} &= \oplus(\mathbf{c_1}, \mathbf{c_2}, \cdots, \mathbf{c_m}) \\ &= \{(u, \mathbf{g}(u)) : \mathbf{g}(u) = (g_1(u), g_2(u), \cdots g_m(u))\}. \end{aligned}$$

where $\oplus$ is the *union* variant of GIS *overlay* operation that yields a new map by combining all involved features in the input maps; and for $j = 1 \cdots m$, $g_j(u)$ is the criterion function associated with criterion map $\mathbf{c_j}$. Intuitively, criteria maps must represent the same territory and must be defined according to the same spatial scale and the same coordinate system. Note also that overlay operation may generate silver polygons which should be eliminated. In addition, we mention that criteria maps must be polygonal ones. However, input datasets may be sample data points, raster maps, contour lines lines, etc. We need to transform all non-polygonal input datasets into polygonal ones. For example, a set of sample points may be transformed into a TIN by computing the triangulation having vertices at data points or contour lines may be transformed into a polygonal map by a constrained Delaunay triangulation (see e.g. [7]).

The first version of $\mathbf{M}$ is then obtained by applying the multicriteria sorting model Γ_ω to associate each spatial unit u in $\mathbf{I}$ to a category in E:

$$\begin{aligned} \mathbf{M} : \mathbf{I} &\longrightarrow E \\ u &\longrightarrow \Gamma_\omega(u) \end{aligned}$$

The multicriteria sorting model Γ_ω used here will be detailed in §4.3. To generate the final decision map $\mathbf{M}$, we need to group, using Algorithm 1 below, the neighbors spatial units which are assigned to the same category. There are different ways to define the "neighbors" concept. Here, we consider that two spatial units u_i and u_j are neighbors only and only if they share at least one segment: $(\partial u_i \cap \partial u_j \neq \emptyset) =$ true. Other neighboring models may also apply. Note that $\upsilon(u)$ in Algorithm 1 denotes the set of "neighbors" of u.

```
Algorithm 1 GROUPING (M)
begin
u ← u_1
Z ← ∅
While (∃u ∈ I ∧ u ∉ Z)
   For each s ∈ υ(u)
      If Γ_ω(s) = Γ_ω(u) Then
         MERGE(u, s)
      End If
   End For
   Z ← Z ∪ {u}
End While
end.
```

MERGE is a map algebra operator permitting to combine two or more spatial units.

4.3 Multicriteria Sorting Model

The multicriteria sorting model Γ_ω used is ELECTRE TRI (see [6]). The levels of scale E represents the evaluations of p categories defined in termes of a set of $p-1$ profiles. Let $B = \{b_1, b_2, \cdots, b_{p-1}\}$ be the set of indices of the profiles, b_h being the upper limit of category C_h and the lower limit of category C_{h+1}, $h = 1, 2, \cdots, p$. Each profile b_h is characterized by its performances $g_j(b_h)$ and its thresholds $p_j(b_h)$ (preference thresholds representing, for two alternatives x and y, the smallest difference compatible with a preference in favor of x in respect to criterion g_j), $q_j(b_h)$ (indifference thresholds representing, for two alternatives x and y, the largest difference preserving an indifference between x and y in respect to criterion g_j) and $v_j(b_h)$ (veto thresholds representing, for two alternatives x and y, the smallest difference $g_j(y) - g_j(x)$ incompatible with xSy).

The preference parameters vector associated with ELECTRE TRI is $\tau = (\mathbf{k}, \mathbf{q}, \mathbf{p}, \mathbf{v}, \mathbf{B})$, where: (i) $\mathbf{k} = (k_1, \cdots, k_m)$ is the weights vector associated with evaluation criteria reflecting their importance for the DM; (ii) $\mathbf{q} = [q_j(b_h)]$, $j \in F$, $h \in B$ is the indifference thresholds parameters; (iii) $\mathbf{p} = [p_j(b_h)]$, $j \in F$, $h \in B$ is the preference thresholds parameters; (iv) $\mathbf{v} = [v_j(b_h)]$, $j \in F$, $h \in B$ is the veto thresholds parameters; and (v) $\mathbf{B}= (\mathbf{b_0}, \mathbf{b_1}, \cdots, \mathbf{b_p})^T$

is the profiles evaluation matrix with $\mathbf{b_h}= (g_1(b_h), \cdots, g_n(b_h))$. Note that $\mathbf{b_0}$ and $\mathbf{b_p}$ are defined as follows: $\mathbf{b_0} = (\min_{u\in U}(g_1(u), \cdots, \min_{u\in U}(g_m(u))$ and $\mathbf{b_p}(\max_{u\in U}(g_1(u), \cdots, \max_{u\in U}(g_m(u))$. It is obvious that different values for w may lead to different assignment results.

ELECTRE TRI has two assignment algorithms: pessimistic and optimistic. Algorithm 2 provides the pessimistic version.

```
Algorithm 2 ASSIGNMENT (Γ_ω(u), ∀u ∈ I)
begin
For each u ∈ I
   h ← p
   g(u) ← (g_1(u), ···, g_m(u))
   assigned ← False
   While h ≥ 0  and  ¬(assigned)
     g(b_h) ← (g_1(b_h), ···, g_m(b_h))
     w' ← (q(b_h), p(b_h), v(b_h))
     If SIGMA(g(u), g(b_h), k, w') ≥ λ Then
        Γ_ω(u) ← e_{h+1}
        assigned ← True
     End If
     h ← h − 1
   End While
End For
end.
```

The boolean variable `assigned` is used to avoid unnecessary loops. The algorithm `SIGMA` permits to compute credibility index $\sigma(u, b_h)$ measuring the degree to which spatial unit u outranks profile b_h: uSb_h. The complexity of `SIGMA` is $O(m)$; where m is the number of evaluation criteria (see [6] for more information). The parameter $\lambda \in [0.5, 1]$ is the cutting level representing the minimum value for $\sigma(u, b_h)$ so that uSb_h holds.

5 Generating Spatial Decision Alternatives

As mentioned earlier, spatial decision alternatives are often modeled through punctual, linear or areal features. This may generate a large set of alternatives which makes outranking methods non practical since they quickly reach their computational limitations. To avoid this problem, we propose in this section different solutions to generate these alternatives based on the decision map concept introduced in §4. The basic idea of our solutions consists in "emulate" punctual, linear and areal decision alternatives through one or several spatial units with some additional topological relationships. Accordingly, *punctual alternatives* are modeled as individual spatial units, *linear alternatives* are modeled as a collection of linearly adjacent spatial units, and *areal alternative* are modeled as a collection of contiguous spatial units.

5.1 Generating Punctual Alternatives

Punctual alternatives apply essentially to location problems. They may be modeled as individual spatial units. Thus, the set of potential alternatives A is simply the set of spatial units. Theoretically, any spatial unit may serve as an alternative. However, in practice the DM may wish to exclude some specific spatial units from consideration. Let $X \subset U$ be the set of excluded spatial units: $X = \{u'_i : u'_i \in U$ and that DM states that $u'_i \notin A\}$. Thus, the set of potential alternatives is: $A = \{a_i : a_i \in U \setminus E\}$.

5.2 Generating Linear Alternatives

Linear alternatives are often used to model linear infrastructures as highways, pipelines, etc. They may be modeled as a collection of linearly adjacent spatial units. The generation of this type is more complex than the punctual ones. In this paper, these alternatives are generated basing on the connexity graph resulting from the decision map. The connexity graph $G = (U, V)$ is defined such that : $U = \{u : u \in \mathbf{M}\}$ and $V = \{(u_i, u_j) : u_i, u_j \in U \wedge \partial u_i \cap \partial u_j \neq \emptyset \wedge u_i^\circ \cap u_j^\circ = \emptyset\}$. Each vertices x in G is associated with the evaluation $v_E(u)$ of the spatial unit u it represents.

In practice, the DM may impose that the linear alternative t must pass through some spatial units or avoid some other ones. Let $Y = \{u \in U : (t \cap u = u) \wedge (t^\circ \cap u^\circ \neq \emptyset)\}$ be the set of spatial units that should be included and $X = \{u \in U : (t \cap u = \emptyset)\}$ be the set of spatial units to be avoided. The conditions in the definition of set Y signify that $(u, \texttt{in}, t)$ is true and the one in the definition of X means that $(u, \texttt{disjoint}, t)$ is true. Let also $\texttt{f}(t)$ and $\texttt{t}(t)$ denote the start and end spatial units for an alternative t. A linear alternative t is defined as follows:

$$t = \{u_1, \cdots, u_q : u_i \in U \setminus X, i = 1..q\}$$

with: (i) $\texttt{f}(t) = u_1$ and $\texttt{t}(t) = u_q$; (ii) $(\partial u_i \cap \partial u_{i+1}) \neq \emptyset \wedge u_i^\circ \cap u_{i+1}^\circ = \emptyset, \forall i = 1..q-1$; and (iii) $t \cap Y = Y$. The first condition set the origin and end spatial units. The second condition ensures that spatial units in t are linearly adjacent. The last one ensures that all spatial units in set Y are included in t. Alternatives are then generated based on $G'(U \setminus X, V')$ with the condition that these alternatives should pass through spatial units $u_j, \forall u_j \in Y$. To generate alternatives, we apply the following idea. A linear alternative is defined above as a collection of linear adjacent spatial units. Let $(v_E(u_1), v_E(u_2), \cdots, v_E(u_q))$ be the set of the evaluations of all spatial units composing t. Then, to evaluate an alternative t we need to map a vector of q evaluations to a vector of k evaluations using a transformation rule φ:

$$\begin{aligned} \varphi : E^q &\rightarrow E' \\ (v_E(u_1), v_E(u_2), \cdots, v_E(u_q)) &\rightarrow (e'_1, e'_2, \cdots, e'_k) \end{aligned}$$

where $E' : [e'_1, e'_2, \cdots, e'_k]$ is an ordinal evaluation scale with $e'_1 \prec e'_2 \prec \cdots \prec e'_k$ (E' can be the same one used in decision map generation). The level e'_i may be

the number of nodes x_j (i.e. spatial unit u_j) such that $v_E(x_j) = e_i$, the area of spatial units u_j evaluated e_i, or any other spatial criterion. Before performing the global evaluation, dominated corridors need to be eliminated from consideration. The *dominance relation* $\triangle$ can not be defined directly on the initial evaluation vector $(v_E(x_o), \cdots, v_E(x_n))$ since alternatives may have different lengths (in terms of the number of spatial units). It can be expressed on the transformed evaluation vector as follows. Let t and t' be two linear alternatives with transformed evaluation vectors $(r_1, r_2, \cdots, r_k)$ and $(r'_1, r'_2, \cdots, r'_k)$, respectively. Then t dominates t', denoted $t \triangle t'$, holds only and only if: $t \triangle t' \Leftrightarrow r_i \succeq r'_i, \forall i = 1..k$ with at least one strict inequality. The global evaluation of an alternative t is $v(t) = \mathbf{\Theta}(r_1, r_2, \cdots, r_k)$ where $\mathbf{\Theta}$ is an aggregation mechanism.

5.3 Generating Areal Alternatives

In several problems, alternatives are often modeled as a collection of contiguous spatial units. To generate this type of alternatives, we use the following idea. Let $T^\alpha = \{u_j \in U : v_E(u_j) = \alpha\}$ be the set of spatial units in U with level α; $\alpha = 1..k$. Let $T_i^\beta = \{u_j \in U : \partial u_i \cap \partial u_j \neq \emptyset \wedge v_E(u_j) = \max_{l \in E \wedge l < \beta} l\}$ be the set of spatial units that are contiguous to u_i and having the best evaluation strictly inferior to α. Next, we construct a tree T defined as follows. To each spatial unit u_i in T^α associate spatial units in $T_i^\beta; \beta \prec \alpha$, as suns. Note that if $|T^k| > 1$, we need to create a hypothetic node r having as suns the spatial units in T^k. Then, an areal alternative a may be constructed as a collection of spatial units in an elementary path starting in a spatial unit in T^k (or r if $|T^k| > 1$) and continues until some conditions (e.g. the total surface of spatial units in the path) are verified.

Let $a = \{s_1, s_2, \cdots, s_z\}$ with $s_1 \in T^k$ an elementary path in T representing an areal alternative. As for linear alternatives, dominated areal alternatives should be eliminated from consideration. Let a and a' be two areal alternatives constructed as mentioned above, then $a \Delta a'$ holds if and only if a' is included in a, i.e.:

$$s_i \in a' \Rightarrow s_i \in a, \forall s_i \in a'.$$

Intuitively, all dominated decision alternatives should be eliminated from consideration.

6 Generating Composed Decision Alternatives

In practice, spatial decision alternatives may be modeled as a combination of two or several atomic decision alternatives. For example, a school partitioning problem may be represented by a set "point-area" composed decision alternatives where points schematize schools while areas represent zones to serve. Spatial problems implying composed alternatives may be addressed as follows. The idea consists in decomposing the problem into a series of subproblems each of which implies only one type of atomic decision alternatives. Considering, for instance,

Table 1. Algorithms for generating composed alternatives

Alternative	P'	L'	Q'
P	Find t as in §5.2 with $\mathtt{f}(t)=P$ and $\mathtt{t}(t)=P'$	Find L' as in §5.2 with $P \in Y$	Find Q' as in §5.3 with $(P, in, Q') = true$
L	Find P' as in §5.1 with (P', in, L) =true with (P', in, L)=true	Find L and L' as in §5.2 with, e.g., $L \cap L' \neq \emptyset$, $\mathtt{f}(L)=\mathtt{t}(L')$	Find L and Q' as in §5.2 and §5.3 with, e.g., $L \cap Q' \neq \emptyset$
Q	Find Q and P' as in §5.3 and §5.1 with, e.g., $\mathbf{M} = Q$	Find Q and L' as in §5.3 and §5.2 with, e.g., $(L', \mathtt{cross}, Q)$ = true	Find Q and Q' as in §5.3 with, e.g., $(Q, \mathtt{overlap}, Q')$ = false

that our initial problem involves composed decision alternatives which are constituted of two atomic alternatives γ_1 and γ_2 that should verify a set of spatial relationships R. This problem can be decomposed into two steps: (i) in the first step we resolve a first sub-problem involving alternatives of type γ_1, and (ii) in the second step we resolve another sub-problem involving alternatives of type γ_2 taking into account the spatial relations in R. We note that the spatial relationships in R may be topological, (e.g. $(\gamma_1, \mathtt{disjoint}, \gamma_2)$ = true), metrical (e.g. $dist(\gamma_1, \gamma_2)$ is less to a given value) or directional (e.g. γ_1 is beside γ_2). Only the first family of relationships is considered in the following.

Table 1 presents the general schema of algorithms that may used to deal with problems implying two atomic decision alternatives. In this table the symbols P and P'; L and L'; and Q and Q' denote punctual, linear and areal decision alternatives, respectively. The symbol t denotes a linear alternative. These algorithms use the solutions proposed in §5 with some topological relationships. We remak that generally in a problem implying two types of alternatives we may start by generating either of which. However, generating alternatives of the first type may restrict the solution space for the second type of alternatives.

The algorithms in Table 1 are straightforward. Here we just comment briefly some of them. Considering for example the P/P' case. This type of problems can be simply considered as corridor generation and may be resolved as in §5.2 with the additional conditions $\mathtt{f}(t)=P$ and $\mathtt{t}(t)=P'$ (or the inverse). The Q/Q' case involves two areal decision alternatives that can apply for the location of two facilities. The school partioning problem mentioned above correspond to Q/P' case.

7 Implementation

7.1 Data Structure

We have used an hierarchical data structure composed of three levels. The highest level corresponds to the decision map **M**. The second level corresponds to the aggregated spatial units. The third level is the individual spatial units. Each element of the second level represents a collection of contiguous spatial units having the same evaluation. To avoid redundancy, global evaluations of elements of the second level are not stored in the database but computed when it is required using the partial evaluations associated with individual spatial units of the third level.

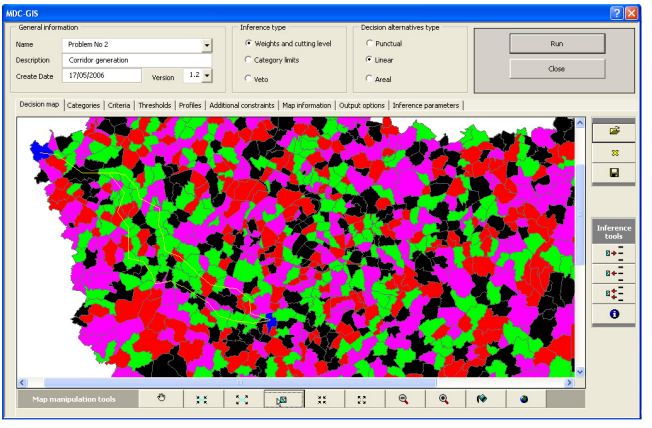


Fig. 1. A screen from the prototype showing two corridors

7.2 Evaluation of Aggregated Spatial Units

The basic elements for the construction of alternatives are the elements of the second level. This requires the definition of appropriate aggregation mechanisms to evaluate each of these elements basing on the evaluations of individual spatial units. Different aggregation schema may be used [2]: (i) statistical operators as summation, maximum and variance may be sued for ordinal or cardinal evaluation criteria. The main advantage of this mechanism is its compactly and simplicity; (ii) functional aggregation mechanisms that are based on the use of a function and works by total or partial aggregation (e.g. weighted sum, integral function). They apply to numerical data; and (iii) rule-based aggregation mechanisms. This type of mechanisms is useful to model complex aggregation logics that are difficult to model within statistic or function aggregation mechanisms.

7.3 Prototype

We have developed a prototype on ArcGIS 9.1 by using VBA. The prototype permits to create criteria maps, infer preference parameters for ELECTRE TRI, assigning spatial units to categories, and generate decision alternatives. We have used real data relative to Ile-de-France (Paris and its suburban) region in France and three illustrative problems of location, corridor generation and zoning have been considered. A full description of the prototype is available in [3]. Here we briefly comment the second problem. Three criteria maps (land-use, sol type and administrative limitations) and four categories have considered. Figure 1 presents two corridors generated using the idea described in §5.2.

8 Conclusion

We have proposed an approach to generate spatial decision alternatives in multicriteria spatial decision making contexts. The proposed approach uses a planar

subdivision of the study area, called decision map, composed of a set of non-overlapping set of spatial units. These spatial units are then used to "emulate" punctual, linear and areal decision alternatives often used as input for MCA methods. This way of modeling permits to reduce significantly the number of alternatives to be evaluated enabling outranking methods to be used. The proposed solutions have been implemented on ArcGIS 9.1 by using real data relative to Ile-de-France region in France.

References

1. Araújo, C., Mecedo, A.: Multicriteria geologic data analysis for mineral favourability mapping: application to a metal sulphide mineralized area. Ribeira Valley Metalogenic Province, Brazil", Nat. Res. Resea. 11, 29–43 (2002)
2. Barrera, R., Egenhofer, M.J., Frank, A.U.: Robust evaluation of spatial queries. In: 5th International Symposium on Spatial Data Handling, Charlestion, South Carolina, August 3-7, 1992, pp. 241–248 (1992)
3. Chakhar, S., Mousseau, V.: MDC-GIS: un SIAD d'aide multicritère à la décision à référence spatiale. In: Septièmes Journées Scientifiques des Jeunes Chercheurs en Génie Electronique et Informatique (GEI 2007), Monastir, Tunisia, Mrach 26-28, vol. 28, pp. 141–150 (2007)
4. Clementini, E., Di Felice, P., van Oosterrom, P.: A small set of formal topological relationships suitable for end-user interaction. In: Ito, T., Abadi, M. (eds.) TACS 1997. LNCS, vol. 1281, pp. 415–438. Springer, Heidelberg (1997)
5. Clementini, E., Di Felice, P.: A comparison of methods for representing topological relationships. Inf. Sci. 3, 149–178 (1995)
6. Figueira, J., Mousseau, V.: B. ELECTRE methods. In: Figueira, J., Greco, S., Ehrgott, M. (eds.) Multiple criteria decision analysis: State of the art surveys, pp. 133–162. Springer, Heidelberg (2005)
7. de Floriani, L., Magillo, P., Puppo, E.: Applications of computational geometry to geographic information systems. In: Sack, J.-R., Urrutia, J. (eds.) Handbook of Computational Geometry, pp. 333–386. Elsevier Science B.V, Amsterdam (1999)
8. Joerin, F., Thériault, M., Musy, A.: Using GIS and outranking multicriteria analysis for land-use suitability assessment. Int. J. Geo. Inf. Sci. 15, 153–174 (2001)
9. Malczewski, J.: A GIS-based multicriteria decision analysis: a survey of the literature. Int. J. Geo. Inf. Sci. 20, 703–726 (2006)
10. Marinoni, O.: A stochastic spatial decision support system based on PROMETHEE. Int. J. Geo. Inf. Sci. 19, 51–68 (2005)
11. Marinoni, O.: A discussion on the computational limitations of outranking methods for land-use suitability assessment. Int. J. Geo. Inf. Sci. 20, 69–87 (2005)

Market-Based Adaptive Discussion Forums*

Natalia López, Manuel Núñez, Pablo Rabanal,
Ismael Rodríguez, and Fernando Rubio

Dept. Sistemas Informáticos y Programación
Facultad de Informática
Universidad Complutense de Madrid
C/. Profesor José García Santesmases s/n
28040 – Madrid. Spain
{natalia,mn,isrodrig,fernando}@sip.ucm.es

Abstract. One of the most successful and simplest Internet-based systems to promote the spread of knowledge are discussion forums. Unfortunately, not much research has been done to improve them by introducing adaptive capabilities. In this paper we discuss a market-oriented mechanism to promote the knowledge exchange activity in discussion forums. Our system provides a strategy to dynamically adapt the structure of the forums by using statistics about users behavior. By doing so, we minimize the response time in which questions are satisfactorily answered. In addition to that, the effort needed to generate useful information is also reduced.

Keywords: E-learning, discussion forums.

1 Introduction

The field of e-learning is reaching a certain degree of maturity. Different aspects are the responsible for this improvement in e-learning. On the one hand, the new technologies and programming tools allow to develop more complex applications. On the other hand, the wide implantation of the Internet has produced an increasing demand for this kind of systems. A notable advance in the creation of intelligent tutoring systems was produced when artificial intelligence techniques appeared in the development of this kind of systems. Good examples of these new generation tutoring systems are [16,7,21,22,14,13,1]. These interactive learning environments allow students to change from passive observers to active constructors of knowledge [2], by favoring *active* learning [9,8]. Besides, by using adaptive hypermedia techniques (e.g. [5,11,23,6,4]) when developing intelligent tutors, we have systems that automatically adapt themselves according to the responses of students (e.g. [21]).

* Research partially supported by the Spanish MCYT project TIN2006-15578-C02-01, the Junta de Castilla-La Mancha project PAC06-0008-6995, and the Marie Curie project MRTN-CT-2003-505121/TAROT.

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 149–160, 2009.

One of the simplest and most classical, but also extremely successful, electronic mechanisms used for e-learning are *discussion forums* and *news groups*. By using these applications, the knowledge about a specific topic can be easily shared. In fact, thanks to the collaborative action of many users, doubts and questions can be solved by their mates, as the more experienced users help beginners by transmitting them their knowledge.

Unfortunately, the cooperation mechanisms appearing in classical discussion forums are still somehow primitive. In fact, they do not usually promote the efficient generation of information. First, the incentives to share information by answering questions of other users can be weak. In most cases, the altruism or the desire of prestige is the main motivation to help beginners. Unfortunately, such motives could be not shared by many users. Thus, the collaboration degree can be quite asymmetric. Second, experts can lose their motivation to help others as they can get saturated by a huge amount of questions. Even though most of the problems could be solved by not so skilled users, more experienced ones receive all the questions.

The key to understand the problem underlying discussion forums is to consider the *generation* of information as a *scarce resource*. More precisely, if we consider the cost of the generation of information in terms of the time needed to create it, then it becomes clear that such activity must be considered a scarce resource. Obviously, the costs to copy, store or transmit information are negligible compared to the cost of generating it. So, we will only consider the aspects related to the generation of information.

Since the efficiency in the generation and use of information in a discussion forum can be defined in terms of optimizing the access to a scarce resource, there exists a science that can help us dealing with it. Quoting from [18], "*Economy is the science that deals with distributions of scarce resources which can be valued by having different utilities by different users.*" In our concrete problem, the effort for generating information is clearly a scarce resource, and the utility that each user of the system gives to each piece of information is also clearly different. As an example, let us consider two different questions performed by two different users. The first of them is more complex than the second one. In fact, only experts can answer such a question, while the second one can be answered by a user with only some command in the topic. In that situation, it is clearly inefficient to ask an expert to answer the easiest question and to assign the hardest question to a user with only some command on the corresponding topic. The most usual method in economic systems to promote the efficient utilization of resources consists in using *prices* for them. These prices define the effort level that an interested agent should apply to obtain a certain resource. If a resource is scarce (supply lower than demand) then the price raises, so only the most interested agents can obtain the desired resource. Moreover, a high price in a resource encourages producers of the resource to increase its production. Thus, the difference in prices among the different resources represents the real necessity of each product as well as promotes the production of the items most estimated by the users.

As it can be expected, the most common unit to measure the effort level to be paid is *money*. However, in order to promote an efficient access to resources it is not necessary to use *real* money: It can be virtual money provided that its quantity is scarce. In fact, several market oriented solutions have already been used to regulate the access to information in databases or the processing time in distributed systems (see e.g. [20,19,17,10,3]). We claim that a market-oriented approach could also be applied by an intelligent autonomous agent to control the generation of information in a discussion forum. In this way, each user should have to *somehow* pay other users in order to get her doubts solved. However, a direct application of the market oriented concepts could not be completely satisfactory, as e-learning environments need to consider additional factors not appearing in other computational systems. In this paper we continue our previous work [15] in order to deal with these important differences and propose a suitable market-oriented methodology to optimize the use of the educational effort in discussion forums. The system has been implemented and our experience indicates that discussion forums may benefit from the ideas presented in this paper.

The structure of the rest of the paper follows. In Section 2 we introduce the main ideas of our market-oriented methodology. Then, in Section 3 we show how discussion forums can be dynamically adapted to improve their efficiency. Afterwards, in Section 4 we present the implementation of the system. Finally, in Section 5 we present our conclusions.

2 Improving the Basic Structure of Discussion Forums

The current section describes how a market-oriented methodology can be applied to optimize the performance in a discussion forum. In particular, we present the main ideas to provide incentives to the users.

2.1 Paying for Good Answers (without Money)

First of all, let us remark again that the utility obtained by receiving the answer to a question (strongly) varies from user to user. Thus, the number of different resources may be, eventually, equal to the number of users, so it can be huge. In this case, the system will not be able to decide, in general, which user is the best choice to answer a concrete question. To overcome this difficulty, the system should dynamically *classify* users based on the *quality* of their effort generating answers. By using classes of users at different levels, the system can determine the suitability of a given user to answer a question of another given user. Let us remark that the quality of the effort of the generation of information does not only depend on the knowledge of the users, but also on their willingness to transmit their knowledge.

It is important to point out that, since we are dealing with knowledge, it is impossible for a user to know with full certainty whether her question requires the help of an expert, or only that of a not so skilled user. Thus, not much improvement of the efficiency could be obtained by allowing the questioner to

choose the type of user that should answer her doubt. On the contrary, the system must provide a mechanism of distribution of questions to decide which type of users should answer each type of questions.

Another important point is the fact that the quality of the resources, that is, the quality of the received answers, is not known a priori. In other words, a user asking a question does not know how good the answer is until she obtains it. In fact, she can get disappointed with an incomplete/erroneous/bad answer. Analogously, a user answering a question cannot know the quality of her answer until the other user studies it. Thus, it is not possible to set the price of an enquiry a priori. Actually, the receiver of the resource should be the one who fixes the price she will pay. Obviously, if money is the scarce resource that enables access to information then the buyer of the resource will try to pay as few as possible, arguing that she was not satisfied with the answer. This problem can be overcome if the paid *money* does not perceptibly modify the future capacity to *buy* new knowledge. In order to do so, the units paid by users will be potentially *inexhaustible*, as we will comment below.

The main reason for including prices in our approach is to give incentives to the providers of knowledge. Thus, when a user gets satisfied with an answer then the provider of the answer should receive an incentive. As the only resource available in the system is the educational effort, such incentive should be given in terms of it. Since the users will not be allowed to choose the type of users to answer their doubts, the system will have to appropriately award incentives to users, both by providing and by restricting the access to educational effort. If a user provides answers that are positively perceived then:

- The system will try to help the user to get her future doubts solved. As she has proved that she has valuable knowledge, her future questions will be shown to more expert users. Thus, she will subsequently obtain satisfactory answers with a higher probability.
- As she has proved that she has valuable knowledge, the system will also try not to show her easy questions in the future. By doing so, she will be able to save her effort for those questions that really require her skills.

In order to obtain such behavior, users will be split into classes according to their education efforts in the past. Each time a user posts a new question, it will be initially shown only to the users belonging to her own class. If after a period of time (depending on the class) the question is still unanswered then it will be forwarded to the next level. If the question is not answered there, it will be forwarded again and so on until reaching the highest level. So, easy questions will be usually answered before they reach the expert level. This is so because, as we remarked before, the user providing an answer to a question may improve her *reputation* by getting into a *better* group.

Note that, by using this mechanism, we encourage that the questions are answered by those users that are the most appropriate to do it: Users with medium command answer the questions of beginners while experts answer both the questions of the medium users and the ones from beginners which could not be solved by medium users after the corresponding delay. That is, questions are

answered by the less skilled users who can correctly answer them. Therefore, the scarce resources of the system (that is, the effort to create valuable information) are exploited in an efficient way, keeping the most valuable resources for the specific questions that really need them.

Let us remark that the underlying idea of this method consists in restricting the access to educational effort while not restricting the access to the information itself. In fact, in our implementation anybody have access to the full repository of questions and answers (moreover, standard SQL queries are available for searching the appropriate information).

2.2 Structuring Users in Classes

Many discussion forums (e.g. [12]) allow users to be structured in different classes. However, this distribution usually depends on the interest of the users on different topics. In our discussion forum, the distribution of users among classes does not depend on their interest on specific topics, but on their previous effort generating information for other users. Each group is a set of sorted users (according to the amount of gained points). So, the set of *guru* users is made of the u_1 best users of the ranking, the set of *expert* users is made of the next u_2 users, and so on. As it usually happens in knowledge communities, the amounts of users in each class should follow a pyramidal structure. Thus, the condition $u_1 < u_2 < \ldots < u_n$ will be considered.

Let us remark that by structuring classes through the ranking system a user can, after some time, either improve or fall in the hierarchy. For instance, in the case that a user reduces her activity, she will be overtaken by other more active users. Therefore, even if the points each user owns to acknowledge other users are inexhaustible, these points can actually affect her. Let us remark that, however, the effect of acknowledging others has a minimal repercussion if we consider them *individually*. In a system with thousands of users, giving points to some other specific user will not have, in general, a perceptible effect on the ranking of the specific user who gives them. Nevertheless, the result of the simultaneous delivering of points of all the users of the system yields a competitive environment as a whole.

Let us note that, by using this method, the incentive for a user to reduce the points she gives to others is very low, because what can actually constrain her access to the resources is mainly the activity of *others*. This is specially clear if we compare this method with a method where the units each user delivers are *exhaustible* and each user is forced to pay fixed amounts of units before accessing the information provided by another user. Let us also note that structuring classes by fixed intervals of points (for example, users in the range 1000–1500 are located in the *medium* class) leads to the same competitive behavior: even if the points cannot decrease and degradation is impossible, quality of the accessed information effort can fall. The reason is that what is really important about being in a given class is not the name of the class but the set of mates one has in the class.

It is important to remark that the distribution of users among classes is dynamically done by taking into account their behavior. However, when creating

a forum for the first time there is not information about the previous behavior of the users. In that case, the administrator of the forum can choose between assigning the same initial level to all the users, or creating some initial classes based on *her* knowledge about the users. Note that, in many situations, the administrator can have information about the skills of some of the users. For example, when creating a new forum on a specific research topic, the set of guru users can be easily obtained by considering the publications of the users about this concrete research topic.

3 Adaptive Capabilities of the System

In this section we briefly comment how to adjust certain parameters in order to minimize the mean time required to satisfactorily solve the questions. The interested reader can find more details in [15]. The main parameters to be dynamically adjusted are the *waiting times*, that is, the times questions need to stay unsolved at each level before forwarding them to an upper level. We will denote by t_{ij} the time that questions of users of level i remain in level j before forwarding them to level $j+1$. Let us remark that these values should be neither *high* nor *low*. On the one hand, if they are very high then the response time will unnecessarily increase: If a question is not solved in a reasonable time then it will probably not be solved in any time. On the other hand, if the time is very low then the probability to receive an answer at each level is very small. Hence, upper levels would receive more questions that needed. In fact, the frequency of questions can saturate the upper levels, reducing the overall efficiency of the system. In this case, the response times would be increased as well. In the rest of this section we comment how to compute appropriate *intermediate* values for the waiting times.

The factors we will use to optimize waiting times t_{ij} will be controlled by users statistics. These data can be automatically obtained by the system without affecting the normal performance of the application. Let us suppose that there are n levels of users in the system. Then, for $1 \leq i \leq n$ we need to compute the frequency f_i of generation of questions of users belonging to level i. Besides, we use a set of random variables to codify the time needed for questions to be correctly answered in a given level. That is, discrete random variables ξ_{ijk} describe the time required for a question coming from users of level i to be solved by users of level j, assuming that the frequency of questions arriving at level j (both from level j and other lower levels) is k. Let us remark that the frequency of incoming questions influences the response time. This is so because a high value implies that users of class j will waste a lot of time just reading the upcoming questions. So, their efficiency answering questions will decay.

By considering the previous data, we can compute the probability p_{ijkt} that the random variable ξ_{ijk} takes a value less than or equal to t, that is, p_{ijkt} will represent the probability that t units of time have passed. So, the following equality holds: $p_{ijkt} = Prob(\xi_{ijk} \leq t) = \sum_{x=0}^{t} f(x)$, where f is the probability density function associated with ξ_{ijk}. In the following we will denote by p_{ijk} the

probability of answering in level j a question initially placed at level i, assuming an input frequency k. That is, p_{ijk} stands for $p_{ijkt_{ij}}$.

By considering the previous random variables, the real frequency of questions arriving at each level can be computed. Basically, the level i will receive those questions generated at that level as well as those questions coming from lower levels that have not been successfully answered in the previous levels. Thus, if we denote by g_i the frequency of questions arriving at level i then we have $g_i = \sum_{1 \leq j \leq i} f_j \cdot \prod_{j \leq k < i} (1 - p_{jkg_k})$. Note that g_i depends on other values g_k, so it is defined in a recursive fashion. This recursion is well founded because when g_i depends on g_k we have $k < i$.

Finally, in order to compute the mean response time for users of level i, we add the mean times needed to obtain the answers at each level from i to $n+1$, weighting this time with the probability that the question is satisfactorily answered at this level. Due to lack of space we do not show the formula computing it, but it can be found in [15]. Then, the problem of deciding the best *waiting times* is reduced to obtaining the best values to optimize the formula.

Let us finally remark that it is also interesting to reduce the variance of response times among different levels. That is, it is interesting to try to obtain similar mean response times in all the levels. One could argue that if the mean response times are the same for all classes then it is not relevant the class a user is located in. So, there is no motivation to answer the questions of others. However, this is false. If the variance is low, it means that the *mean* response times are similar for all classes. In each class, the considered response time is the mean time for all the members of the class. So, if a user inside a class learns and improves her skills, there will be a time when her questions will be harder to be answered. As a whole, the mean response times of the class could remain the same, but her *own* response times could grow. So, the only way for this user to keep her response times will be to pass to the next class. Therefore, the incentives for users to answer questions remain even if the variance is low. That is, a reduction of the variance of response times does not eliminate the incentives for users to improve their classes.

4 Implementation Details

In this section we describe our implementation of the system. In particular, we will comment on the technologies we have used during the development of the system. Moreover, we will also describe the different views available in the system. First, let us consider the technologies used to construct the implementation. In order to improve the portability and reusability of the system, open source systems have been used as much as possible. More specifically, we have used the following:

PHP: Most web pages of the system have been implemented by using this interpreted language.

Apache: Our server application uses it to serve the web pages.

MySQL: All the information available in the discussion forum is managed by using it.

Fig. 1. User point of view of the forum

QK SMTP Server: We have used it as smtp server to send e-mails directly from the local host to the mailboxes of the receivers.

JavaScript: This programming language - interpreted by the browser - was used as complement of PHP to implement certain peculiarities of the discussion forum.

After introducing the basic technology used in the implementation, now we will concentrate on the views provided to the user. Thus, now we briefly present the main screens of the application interface. Moreover, we will outline the behavior of the hierarchical forum. First, let us focus on the view shown to normal users. Before granting users with the capability of posting messages, a welcome screen requests for a registered user name to enter the forum. As usual, a login and a password are required. If the user does not posses a registered user then she may check in at the system by accessing the *Get Registered* functionality. As it is usual, we also offer the possibility to *Remember Password*. Let us remark that a help system is always available for the user. In particular, it explains how to create a new account, how to solve problems with a previously created account, etc.

Once within the forum (see Figure 1), the user can change the personal data associated with her account, or she can actually uses the forum. In this case, she can access the themes exposed on the forum by pressing on any of the titles exposed in the tree. She can also search for specific topics (see Figure 2), create new questions, answer previously created questions, etc. It is also possible to sort the messages according to desired criteria (date, level, author, etc.). Besides, the user can search messages by using either fast queries (by title, author or contents) or advanced queries. In addition to the previous options, the user can also request to receive an e-mail every time an answer to a specific topic is posted in the forum. By doing so, we provide an alternative way to check the status of a given question.

Now we focus on the application view for the administrator of the system. Although the system can adapt itself automatically, we also allow the

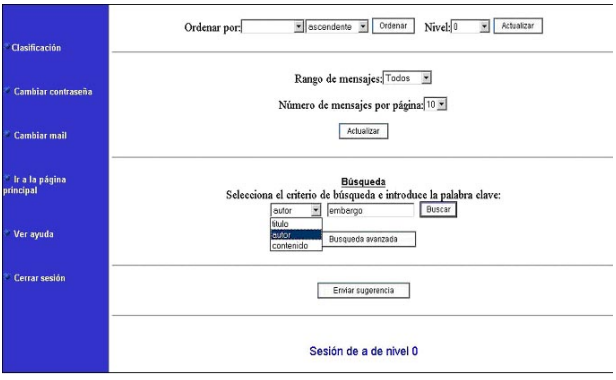


Fig. 2. Searching messages in the forum

administrator to monitor the evolution of the system. In particular, she receives alerts each time a relevant change happens in the system. The available options include reading the help system, removing users and discussion threads (to protect the system against illegal or undesired uses), adjusting system variables, and obtaining statistics about the use of the system. Let us briefly describe the most relevant features:

Queries. The most typical queries needed by the administrator are predefined (questions accomplished in the later month; registered users; percentage of answers not punctuated yet; requesting consultations on given punctuations; etc.) Moreover, the system also allows to introduce any SQL query against the database. By doing so, we provide a great flexibility to the administrator, as she can access any information available in the system.

Graphics. The system also allows the generation of special graphics to improve the readability of certain statistics. In particular, predefined special graphics include the following: Number of users per level (see Figure 3, left);

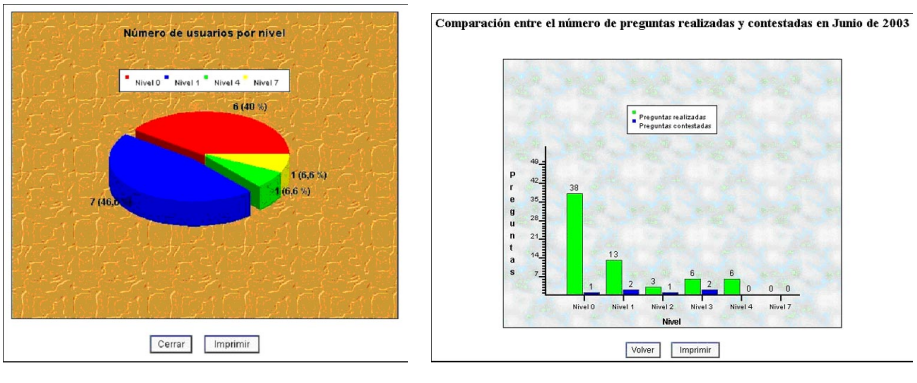


Fig. 3. Graphical statistics provided by the system

number of messages per level; average days to punctuate a question per level; comparison between the questions and the answered questions (see Figure 3, right); average number of days to answer questions; and number of days to answer each question.

Help. In addition to the standard help pages of the system, the system also provides help on the structure of the database's tables. Let us remark that this information is very important to allow the administrator to use SQL queries against the system database: In case the administrator has no information about the internal structure of the database, she cannot take profit of all the advanced queries available by using SQL queries.

5 Conclusions

A market-oriented implementation has been presented to improve the performance of discussion forums. The main idea underlying our tool is that questions should be answered by the less qualified users who are still able to solve the problem. By doing so, the effort of more experienced users is saved, so that they can use it for answering those questions that really require their advance knowledge. In order to achieve this objective, users are dynamically distributed in groups according to the capabilities that they have shown by answering the questions of other users. If a user has a new question then she initially posts it in her group. If after a certain amount of time nobody answers the question then the question is moved to a higher-level group. Let us remark that the system dynamically changes the group of each user by taking into account the perceived quality of their answers.

We would like to point out that we have already provided an implementation of a discussion forum based on the ideas presented in the paper. In fact, in the previous section we have commented some of the characteristics of the system we have implemented. In particular, we have described both the technologies we have used and the main functionalities we provide.

Acknowledgments

The authors would like to thank Javier Fraile and Tomás Muñoz for their contribution in the implementation of the system. The authors would also like to thank the anonymous referees for valuable comments on previous versions of the paper.

References

1. Aïmeur, E., Onana, F.S.M., Saleman, A.: Sprits: Secure pedagogical resources in intelligent tutoring systems. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 237–247. Springer, Heidelberg (2006)
2. Angelo, T.: The campus as learning community: Seven promising shifts and seven powerful levers. AAHE Bulletin 49(9), 3–6 (1997)

3. Ardaiz, O., Artigas, P., Eymann, T., Freitag, F., Navarro, L., Reinicke, M.: The catallaxy approach for decentralized economic-based allocation in grid resource and service markets. Applied Intelligence 25(2), 131–145 (2006)
4. Bra, P.D., Aerts, A., Berden, B., de Lange, B., Rousseau, B., Santic, T., Smits, D., Stash, N.: AHA! the adaptive hypermedia architecture. In: HYPERTEXT 2003: Proceedings of the fourteenth ACM conference on Hypertext and hypermedia, pp. 81–84. ACM Press, New York (2003)
5. Brusilovsky, P.: Efficient techniques for adaptive hypermedia. User Modeling and User-Adapted Interaction 6(2-3), 87–129 (1996)
6. Brusilovsky, P., Maybury, M.T.: From adaptive hypermedia to the adaptive web. Communications ACM 45(5), 30–33 (2002)
7. Brusilovsky, P., Schwarz, E., Weber, G.: ELM-ART: An intelligent tutoring system on the World Wide Web. In: Lesgold, A.M., Frasson, C., Gauthier, G. (eds.) ITS 1996. LNCS, vol. 1086, pp. 261–269. Springer, Heidelberg (1996)
8. Carver, R.H.C., Lane, W.: Enhancing student learning through hypermedia courseware and incorporation of student learning styles. IEEE Transactions on Education 42(1), 33–38 (1999)
9. Davidovic, A., Trichina, E.: Open learning environment and instruction system (OLEIS). SIGCSE Bulletin 30(3), 69–73 (1998)
10. Eymann, T., Reinickke, M., Ardaiz, O., Artigas, P., Freitag, F., Navarro, L.: Self-organizing resource allocation for autonomic network. In: Database and Expert Systems Applications, 2003, pp. 656–660 (2003)
11. Ferrandino, S., Negro, A., Scarano, V.: CHEOPS: Adaptive hypermedia on World Wide Web. In: Steinmetz, R. (ed.) IDMS 1997. LNCS, vol. 1309, pp. 210–219. Springer, Heidelberg (1997)
12. Greer, J., McCalla, G., Vassileva, J., Deters, R., Bull, S., Kettel, L.: Lessons learned in deploying a multi-agent learning support system: The I-Help experience. Artificial Intelligence in Education, 410–421 (2001)
13. López, N., Núñez, M., Rodríguez, I., Rubio, F.: Including Malicious Agents into a Collaborative Learning Environment. In: Cerri, S.A., Gouardéres, G., Paraguaçu, F. (eds.) ITS 2002. LNCS, vol. 2363, pp. 51–60. Springer, Heidelberg (2002)
14. López, N., Núñez, M., Rodríguez, I., Rubio, F.: WHAT: Web-based haskell adaptive tutor. In: Scott, D. (ed.) AIMSA 2002. LNCS, vol. 2443, pp. 71–80. Springer, Heidelberg (2002)
15. López, N., Núñez, M., Rodríguez, I., Rubio, F.: Encouraging knowledge exchange in discussion forums by market-oriented mechanisms. In: SAC 2004, pp. 952–956. ACM Press, New York (2004)
16. McCalla, G.: The search for adaptability, flexibility, and individualization: Approaches to curriculum in intelligent tutoring systems. In: Jones, M., Winne, P. (eds.) Foundations and Frontiers of Adaptive Learning Environments, pp. 91–122. Springer, Heidelberg (1992)
17. Miller, M., Drexler, K.: Markets and computation: Agoric open systems (2000), `http://www.agorics.com/agoricpapers.html`
18. Robbins, L.: An Essay on the Nature and Significance of Economic Science. Macmillan, Basingstoke (1984) (originally published in 1932)
19. Stonebraker, M., Aoki, P., Litwin, W., Pfeffer, A., Sah, A., Sidell, J., Staelin, C., Yu, A.: MARIPOSA: a wide-area distributed database system. The VLDB Journal 5, 48–63 (1996)
20. Waldspurger, C., Hogg, T., Huberman, B., Kephart, J., Stornetta, S.: Spawn: A distributed computational economy. IEEE Transactions on Software Engineering 18, 103–117 (1992)

21. Weber, G.: Adaptive learning systems in the World Wide Web. In: 7th Int. Conf. on User Modelling, UM 1999, pp. 371–378. Springer, Heidelberg (1999)
22. Woolf, B., Beck, J., Elliot, C., Stern, M.: Growth and maturity of intelligent tutoring systems: A status report. In: Forbus, K., Feltovich, P. (eds.) Smart Machines in Education, ch. 4, AAAI Press/The MIT Press (2001)
23. Wu, H., De Bra, P.M.E., Aerts, A., Houben, G.-J.: Adaptation control in adaptive hypermedia systems. In: Brusilovsky, P., Stock, O., Strapparava, C. (eds.) AH 2000. LNCS, vol. 1892, pp. 250–259. Springer, Heidelberg (2000)

Keyword Enhanced Web Structure Mining for Business Intelligence

Liwen Vaughan[1] and Justin You[2]

[1] Faculty of Information and Media Studies
University of Western Ontario
London, Ontario, N6A 5B7, Canada
lvaughan@uwo.ca
[2] ApacBridge Consulting
8 Northgate Street
Ottawa, Ontario, K2G 6C7, Canada
justin.you@apacbridge.com

Abstract. The study proposed the method of keyword enhanced Web structure mining which combines the ideas of Web content mining with Web structure mining. The method was used to mine data on business competition among a group of DSLAM companies. Specifically, the keyword DSLAM was incorporated into queries that searched for co-links between pairs of company Websites. The resulting co-link matrix was analyzed using multidimensional scaling (MDS) to map business competition positions. The study shows that the proposed method improves upon the previous method of Web structure mining alone by producing a more accurate map of business competition in the DSLAM sector.

Keywords: Web content mining, Web structure mining, E-commerce, business intelligence.

1 Introduction

Web data mining can be classified into the following three sub-areas based on the type of data used [1]: Web content mining, Web structure mining, and Web usage mining. Web content mining uses Web page content, the most common of which is Web page texts. Web structure mining tries to discover the model underlying the Web hyperlink structure. Web usage mining tries to discover patterns from Web usage data [2]. Web data mining has been applied to various areas; one of which is E-commerce or business information [3]. Earlier studies used Web content mining [4] and Web structure mining [5] to gather information on business competition, an important topic of business intelligence. Building on to earlier studies, the current study proposes a method that combines both Web content mining and Web structure mining. The method was tested in the Websites of a group of companies in DSLAM (Digital Subscriber Line Access Multiplexer) sector of the telecommunication industry. The result shows that the proposed method of keyword enhanced Web structure mining improves upon the method that is based on Web structure mining alone.

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 161–168, 2009.

We need to define some terms before discussing the details of the study. Inlinks (also called back links) are links coming into (or pointing to) a Web page while outlinks are links going out from a Web page, i.e the hyperlinks embedded in the Web page. Two different types of inlinks need to be distinguished, total inlinks and external inlinks. Total inlinks include all links pointing to a particular page or site while external inlinks include only links coming from Websites outside the site in question. In other words, external inlinks do not include links within the site itself, such as the "back to home" type of navigational links within the site. Björneborn & Ingwersen [6] further distinguished between co-inlinks and co-outlinks. If page X and page Y are both linked to by page Z (i.e. page X and page Y both have inlinks from page Z), then X and Y are co-inlinked. The method proposed in this paper analyzed patterns of co-inlinks (also called co-links later in the paper). Our study only examines co-links in the form of external inlinks because they are objective measures of relatedness between the co-linked sites while internal in-links do not indicate such relationships.

2 Problem Statement

In an earlier study [5], co-link method was used to analyse business competition among a group of companies in the overall telecommunication area. The study showed that this method was able to successfully cluster the competitors based on their overall business strength, market segments and regional market focus. The value of this method is to reveal competitive positions at a macro level.

In a particular product or market segment, for example DSLAM products which are used for DSL broadband Internet access, there are various types of companies ranging from big telecommunication equipment companies such as Ericsson and Alcatel which have many product lines in addition to DSLAM products to small companies which solely focus on one particular product line. Using Web structure mining alone will result in an unbalanced comparison between big companies, which have rich portfolio of products and much wider market presence, and small companies, which specialize in certain products. The Websites of these large companies attract huge number of inlinks and co-links, many of which are related to product lines other than the one being analyzed. In other words, inlink or co-link analysis alone will not result in an accurate comparison among these companies in a particular sector.

To solve this problem, we propose a "keyword enhanced Web structure mining" method in this paper. DSLAM sector is chosen as a case study of a particular market segment. The purpose of the study is to find out whether the proposed method will provide a more accurate competitive analysis for a particular product or a market segment.

3 Proposed Method

The proposed method is based on the idea that co-links to a pair of Websites is an indicator that the two sites are similar or related. The more such co-links, the stronger the relationship. In business world, co-links are particularly useful as business competitors tend not to link to each other so simple in-links do not contain much useful information on business competition. However, two related companies will be co-linked by a third party such as a customer or a reseller [7]. The more co-links the two companies have, the more closely related they are. Since related companies are

competing companies, Web co-link data can be used to cluster companies into a map of business competition. Using this method, an earlier study [5] successfully generated a map of business competition for a group of companies in the telecommunication equipment industry. The current method improves upon the previous method by incorporating Website content, specifically keywords on the sites, to achieve a more accurate mapping.

The method was tested in a group companies in DSLAM sector of the telecommunication industry. DSLAM was chosen for this study as the acronym DSLAM is usually used in the industry and on Websites instead of the complete spelling of the term. So the unique acronym is ideal to test the proposed method as incorporating this keyword into co-link data collection will filter out Websites that co-linked two companies for reasons other than DSLAM, e.g. two sites were co-linked because of their charity activities. We selected 35 DSLAM companies that are included in a reliable research report on DSLAM market [8]. These 35 companies are major DSLAM product companies worldwide.

We located Websites of these companies and then searched for co-links to a pair of these companies using search engine Yahoo! (search details below). The keyword DSLAM is added in search queries. The query result is a matrix of 35 by 35 symmetrical by the diagonal. Each row or column represents a company. Each cell of the matrix records the number of Web pages retrieved by the co-link query. The matrix is not sparse as there was at least one co-link between the majority pairs of companies. This raw co-link matrix needs to be normalized to obtain a relative measure of the strength of the relationship because a co-link count of 5 is very high if the number of links pointing to each company is 6 while it will be low if the number of links pointing to each company is 100. The normalization is done through Jaccard Index as follows:

$$NormalizedColinkCount = n(A \cap B)/n(A \cup B)$$

Where A is the set of Web pages which links to company X
B is the set of Web pages which links to company Y
$n(A \cap B)$ is the number of pages which link to both company X and company Y, i.e. the raw co-link count
$n(A \cup B)$ is the number of pages which link to either company X or company Y.

Multidimensional Scaling (MDS), a statistical analysis method, was then applied to the normalized co-link matrix using version 12 of SPSS software. The MDS output includes a map that positions each company according to their similarity to other companies as measured by co-link counts. The higher the co-link count, the closer the two companies will be placed. Essentially the map will cluster competing companies together so the map will show the competition landscape of the DSLAM sector.

4 Data Collection Details

Yahoo! was used for data collection as it is more suitable for the study than two other major search engines in the market, Google and MSN. As explained earlier, the study needs to search for external inlinks but Google can only search for total inlinks, i.e. it cannot filter out internal links in search results. In order to filter out internal links, the

link query needs to be combined with the "site" query. However, Google's link query term cannot be combined with other query terms as is stated in Google search API reference [9]. MSN can search for external inlinks. However, at the time of data collection (summer 2006), MSN indexed a much smaller number of inlinks than Yahoo! did as MSN usually retrieved a much smaller number of pages for the same inlink query. So Yahoo! was preferred over MSN for data collection.

Two sets of data were collected. One set is the co-link count alone and the other set added the keyword DSLAM in search queries. MDS mapping results from the two data sets were compared to determine if the one with the keyword (i.e. combining link structure data with keyword data) is better. The syntax of queries used to collect the two sets of data is shown in Table 1 in a hypothetical scenario of searching for co-links between www.abc.com and www.xyz.com. Note that Yahoo!, like other major search engines, adds Boolean operator AND by default in between query terms so the AND operator is omitted. In other words, the query syntax for the data without the keyword is effectively "(link:http://www.abc.com –site:abc.com) AND (link:http://www.xyz.com –site:xyz.com)".

Table 1. Yahoo! Query Syntax

Type of Data Collected	Yahoo! Query
Without keyword	(link:http://www.abc.com –site:abc.com) (link:http://www.xyz.com –site:xyz.com)
With keyword DSLAM	((link:http://www.abc.com –site:abc.com) (link:http://www.xyz.com –site:xyz.com)) DSLAM

The "link" command of Yahoo! finds Web pages that link to a particular URL (in this study, links to a company homepage rather than all pages of the company Website). The "linkdoamin" command of Yahoo! will search for Web pages that link to all pages of a site. We decided to use the link command as earlier testing [5] showed that data collected using this command generated better mapping result than that using the "linkdomain" command. A content analysis study [10] that examined reasons of co-linking found that links to homepage were more likely to be business related than links to non-homepage. This confirms that the "link" command is better than the "linkdomain" command for business Websites.

5 Results

Fig. 1 is the MDS mapping result for the set of data that were collected using co-link queries alone (i.e. the first search query in Table 1) while Fig. 2 is the result for the set of data that combined the keyword DSLAM with co-link search queries (i.e. the second search query in Table 1). One needs to be knowledgeable about the industry to interpret the two Figures. The second author has over a decade of experience working in the telecommunication industry. In the interpretation below, the notation of "business competition" is used based on the author's knowledge of the competitive environment of DSLAM industry rather than a strict definition of what constitutes a business competition.

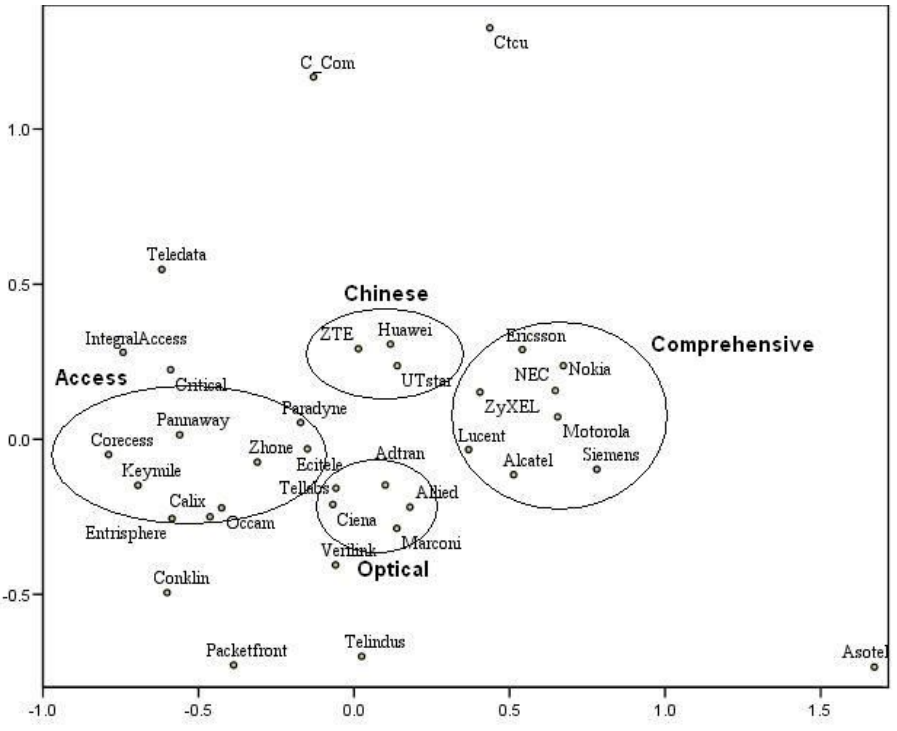


Fig. 1. MDS map without keyword

In Fig. 1, company positions are determined by their overall business and Internet marketing strength in the general telecommunication equipment industry, not specifically in DSLAM marketplace. In this map, tier one telecommunication equipment companies are clustered together in the "comprehensive" category. This group includes Alcatel, Ericsson, and Nokia who are major players in various sectors of the telecommunication industry. The only exception in this group is ZyXEL. ZyXEL is a relatively smaller Taiwanese company with US $325 million revenue a year. However, its strong Internet presence as shown in the Internet Archive (www.archive.org) makes it appear in tier one group. Tier two companies are grouped into several categories including "optical" and "access" companies. The former category includes Ciena, Tellabs, and Marconi which have strong market presence in optical networking while the latter includes Zhone, Paradyne, and Calix, companies that specialize in broadband access products including DSLAM. The Chinese companies are positioned together as a separate group. This reflects the fact that in the general telecommunication equipment market, these companies' main market focus and main competition are mainly in China rather than in Europe and North American. Other small companies are scattered all over the map.

By introducing the keyword DSLAM in search queries (see Table 1), we added a constraint to the relationships among these companies. The effect of this constraint can be seen from the strong contrast between Fig. 1 and Fig. 2. The two Figures are significantly different in that Fig. 1 shows company positions in the general

telecommunication market while Fig. 2 shows their specific market positions in DSLAM sector. In Fig. 2, key DSLAM vendors, such as Paradyne, Zhone, and Calix who are tier two companies in Fig. 1, joined tier one companies to form the center of DSLAM sector (see the inner circle in the middle of Fig. 2). This cluster is denser than any cluster in Fig. 1. The close proximities of these companies in Fig. 2 reflect a very competitive DSLAM market. Other companies took different positions in Fig. 2 as well to form the second tier of DSLAM market as shown by the outer circle in Fig. 2. For example, the Chinese companies do not exist as a separate group in Fig. 2 anymore. Huawei and UTStarcom, the two leading DSLAM equipment companies in the Chinese market, are positioned fairly close to the center of Fig. 2. This reflects the competitive positions of these two companies in the world DSLAM market. In summary, adding the keyword constraint in data collection does serve the purpose of providing a more accurate competitive analysis for a particular market segment (in this case the DSLAM sector).

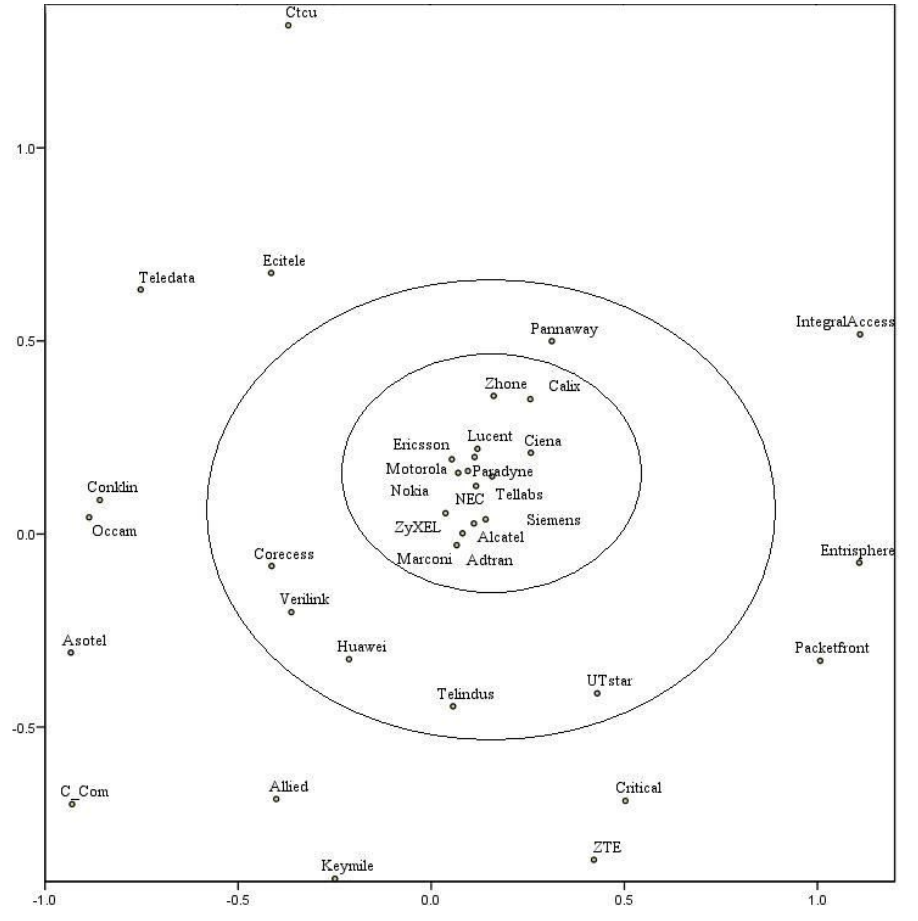


Fig. 2. MDS map with keyword DSLAM

6 Conclusions and Discussion

The study proposed the method of "keyword enhanced Web structure mining" which combines the ideas of Web content mining (keyword) with Web structure mining (hyperlink). The method was used to mine data on business competition among a group of DSLAM companies. Specifically, the keyword DSLAM was incorporated into queries that searched for co-links between pairs of company Websites. Two sets of data were collected: one with the proposed method and one with co-link search alone. The resulting two data matrices were analyzed using multidimensional scaling (MDS) to generate maps of business competition. The comparison between the two maps shows that the proposed method produced a more accurate map of the business competition in the DSLAM sector. However, the proposed method does not refute the previous method of Web structure mining alone. The two methods are suited for different purposes of business intelligence. The method of structure mining alone is suitable for macro-level analysis of business competition as it generates a map of overall competition of an industry. The proposed method is better for micro-level analysis because it produces a map of business competition of a specific sector or segment. This new method represents a level of business intelligence that is higher than what was achieved before.

This is an exploratory study. The method was tested successfully only in one business sector. More testing in other environments is needed to determine if the method can be generalized. The scalability of the method needs to be examined as well. The study is a very first step in combining Web content mining with Web structure mining in that only one keyword was used. Future study will make more sophisticated use of keywords to further improve the method.

Acknowledgements. This study is part of a larger project funded by the Initiative on the New Economy (INE) Research Grants program of the Social Sciences and Humanities Research Council of Canada (SSHRC). Research assistant Karl Fast helped with the programming work for data collection.

References

1. Madria, S.K., Bhowmick, S.S., Ng, W.-K., Lim, E.-P.: Research issues in web data mining. In: Mohania, M., Tjoa, A.M. (eds.) DaWaK 1999. LNCS, vol. 1676, pp. 303–312. Springer, Heidelberg (1999)
2. Lu, Z., Yao, Y., Zhong, N.: Web Log Mining. In: Zhong, N., Liu, J., Yao, Y. (eds.) Web Intelligence, pp. 173–194. Springer, Berlin (2003)
3. Thuraisingham, B.: Web Data Mining and Applications in Business Intelligence and Counter-Terrorism. CRC Press, Boca Raton (2003)
4. Liu, B., Ma, Y., Yu, P.S.: Discovering Unexpected Information from Your Competitors' Web Sites. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, U.S.A, August 26-29 (2001), `http://www.cs.buffalo.edu/~sbraynov/seminar/unexpected_information.pdf`

5. Vaughan, L., You, J.: Mining Web hyperlink data for business information: The case of telecommunications equipment companies. In: Proceedings of the First IEEE International Conference on Signal-Image Technology and Internet-Based Systems, Yaoundé, Cameroon, November 27–December 1, pp. 190–195 (2005)
6. Björneborn, L., Ingwersen, P.: Towards a basic framework of webometrics. Journal of the American Society for Information Science and Technology 55(14), 1216–1227 (2004)
7. Vaughan, L., Gao, Y., Kipp, M.: Why are hyperlinks to business Websites created? A content analysis. Scientometrics 67(2), 291–300 (2006)
8. Beniston, G.: IP DSLAMs, A Heavy Reading Competitive Analysis. Heavy Reading report series 3(15) (August 2005), http://www.heavyreading.com/details.asp?sku_id=836&skuitem_itemid=793&promo_code=&aff_code=&next_url=%2Flist%2Easp%3Fpage%5Ftype%3Dall%5Freports
9. Google (2006). Google SOAP Search API Reference (Retrieved August 18, 2006), http://www.google.com/apis/reference.html#2_2
10. Vaughan, L., Kipp, M.E.I., Gao, Y.: Are colinked business Web Sites really related? A qualitative study. Paper under review in Online Information Review (2006)

Modeling Multi-agent System of Management Road Transport: Tasks Planning and Negotiation

A. Elfazziki, A. Nejeoui, and M. Sadgal

Depart. Computing Sciences, Faculty of Sciences Semlalia, university Cadi Ayyad Bd Prince My Abdellah, BP.2390, Marrakech
{a.elfazziki,a.nejeoui,m.sadgal}@ucam.ac.ma

Abstract. The Management Systems of Road Transport (MSRT) must include responsibility for planning the routes and schedules of vehicles fleet involved in the road haulage, distribution and logistics. It must ensure that all operations are carried out in maximum safety, environmental controls and traffic congestion, driver hours, customs requirements, and minimum cost. The complexity of the real-time scheduling of transport orders which comes in an asynchronous and dynamic way makes the MSRT especially suitable for using techniques from Distributed AI. To manage this complex field addressed under a high degree of dynamism and uncertainty which is characterized by an inherent distribution of knowledge and control, we propose in this work, a modeling of an MSRT by a multi-agents system, the modeling of the agents and their interaction by AUML language and we deal with the cooperation during tasks planning in the MSRT.

Keywords: Multi-agents system, modeling MSRT, Agent Modeling, communication, negotiation, AUML.

1 Introduction

Purely road-based Freight transport has immensely grown during the last decade and is expected to increase even more in the next future due to the drawbacks of Globalization [1]. Additional major contributing factors to this phenomenon are the current production and distribution practices based on low inventories and timely deliveries. Consequently, the development of the new MSRT which integrate planning, optimization and control of transport operations constitutes now one of the most attractive fields for researchers in Distributed Artificial Intelligence (DAI) and transport [2]

In this work, we are initially interested in modeling the MSRT using a multi-agents infrastructure while deploying the techniques of cooperation [3], negotiation, tasks decomposition and tasks allocation, competition and distributed planning in order to enable the MSRT to manage the orders that arrive in an asynchronous and dynamic way and to react in real time to the unforeseen events which can occur during the achievement of the orders. In the second place we tackle the agents modeling using the AUML modeling language. In the third place we illustrate the benefits added by the cooperation among companies during the tasks planning process.

The reminder of this paper is organized as follows. The second section gives a detailed description of the various components of our multi-agents system. The third

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 169–179, 2009.

section presents the various diagrams built using the specifications of the agent unified modeling language (AUML). In section 4 we define the communication protocols and negotiation between the various agents of the MSRT. Section 5 presents static and dynamic tasks planning. In section six we illustrate the benefits added by the cooperation among companies during the tasks planning through an application. Finally, we complete this paper by a conclusion and the prospects for this work.

2 Modeling the Various MSRT Components by Agents

In our approach, we consider the MSRT as a set of companies, vehicles, providers and customers. The goal of each company is to distribute a set of orders on a group of customers in an asynchronous and dynamic way while optimizing its benefits, which explains the competitive aspect among different companies.

The main purpose of our work is to build a multi-agents infrastructure based on agents classification to manage different agent Subsystems at the same time, while keeping a global structure. The MSRT will be composed of the five following subsystems:

A Supervision subsystem made up of a supervisory agent. It deals with the supervision of the whole system.

A Planning subsystem made up of a set of entities (planner agents). Each of these agents will be responsible for freight management, calculation of round and planning of vehicle loading/unloading.

A Customer subsystem which includes all customers. It deals with the management of the various transport orders.

A Providing subsystem made up of a set of provider agents. It manages the various transport services

An Ergonomic subsystem made up of a set of entities (Ergonomic agents). Each one of these agents will be responsible for management of tasks and vehicle-driver assistance.

Therefore an MSRT will be modeled by a Multi-Agents System of the Management of Road Transport (MASMRT). In the following section we present the various agents components of an MASMRT.

Provider Agent: The Provider Agent (PA) can represent a manufacturing unit, a depot of finished products or the distribution department of a company, its principal task is to grant transport orders to the Supervisory agent.

Supervisory Agent: The Supervisory Agent (SA) represents a transport company, after having received a transport order on behalf of a PRA, it splits it into sub-orders which will be the subject of negotiation with its Planner Agents.

Planner Agent: The PLaner Agent (PLA) represents a conveying vehicle with its driver; its principal task is the development of its local plans.

Ergonomic Agent: Responsible for the implementation of knowledge relating to a vehicle, necessary routes, material infrastructures and software available in order to ensure that the missions of transport can be accomplished by the maximum comfort, safety and effectiveness.

Customer Agent: The Customer Agent (CA) represents the final destination of the distributed goods.

3 Modeling of the Agents of a SGTR by AUML

The agent unified modeling language (AUML) (Odell and Al, 2000) was developed by the technical committee of modeling of FIPA (Foundation for Intelligent Physical Agents) like a new methodology able to treat the drawbacks of the functional and object oriented modeling paradigms and to build models based on the concept of the agent as the basic concept.

3.1 Agent Diagram

The first level of AUML modeling defines the classes of agents available in MASMRT and the functionalities they provide on the communicative or actuating actions level. Figure 1 represents a UML diagram that details the functions of the different agents on a "programming" point of view along with the class hierarchy that highlights the polymorphism used for the implementation of these agents. The FIPA Generic Agent, on top of the diagram, defines the basic functionalities an agent must implement to be used as part of the MASMRT platform, which are basically registering agents within the FIPA-OS platform. It demonstrates the steps required in creating an agent that can be instantiated and register with the local MASMRT platform.

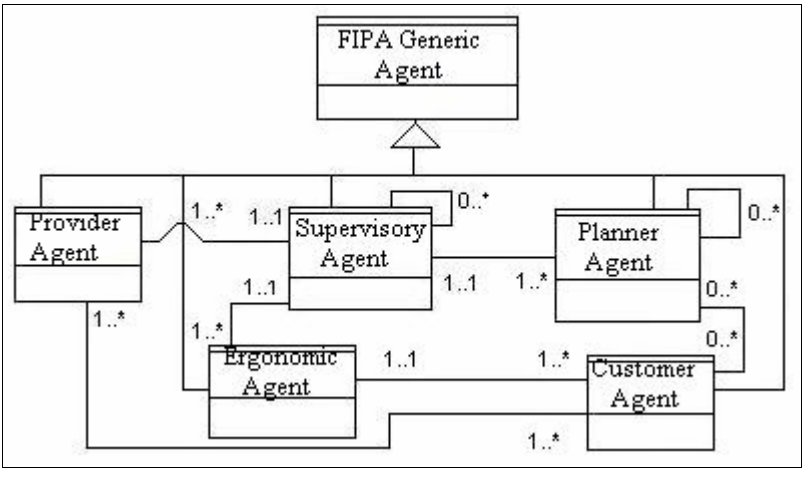


Fig. 1. Agent Diagram of MSRT

To provide a representation which is more appropriate than Figure 1, we have detailed our agents functionalities and behaviors using an AUML *agent diagram*, which has been proposed by Bauer in 2001 (Bauer 01). Each agent is represented in a rectangle divided into five compartments. Figures 2, describe the modeling of the Supervisory Agent. The first compartment contains the agent identifier preceded by the stereotype <<agent>>. The second compartment describes the roles played by the agent. The last compartment deals with the organizations. It defines in which organizations the agent is involved. The actions which the agent can carry out on its environment are represented in the third compartment; the other functions listed in the

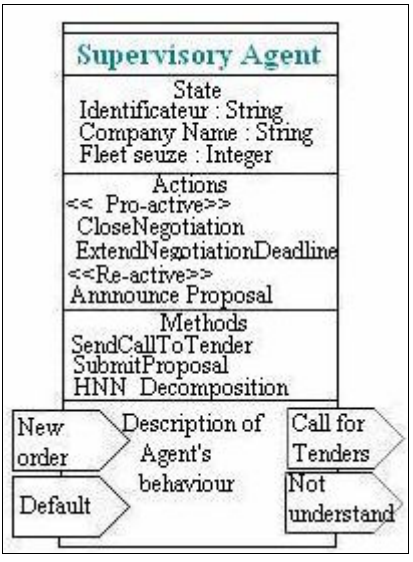


Fig. 2. Supervisory Agent

fourth compartment are standards Java methods. The last compartment represents the library of the communicative acts that the agent can carry out as parts of specific protocols.

3.2 State/Transition Diagram

A state / transition diagram is an automat of finished states which expresses the various states of an agent. A state corresponds to the state of the agent and it includes one or more activities which the agent must carry out. In the following we present the state / transition diagrams for the supervisory agent and the planning agent (cf figures 3 and 4).

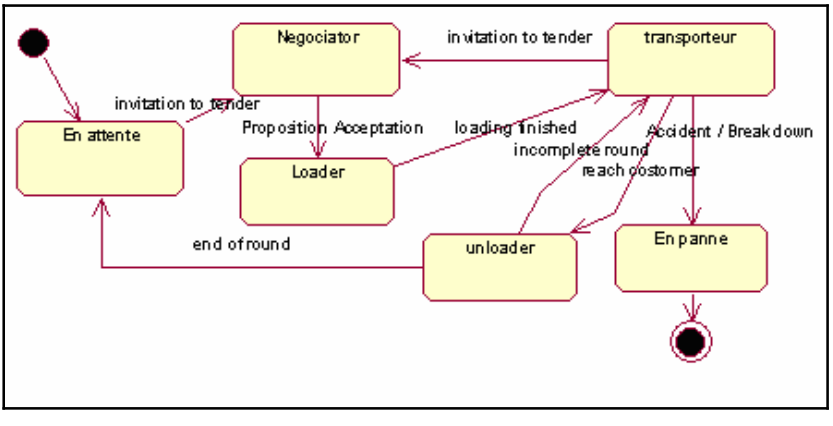


Fig. 3. State/Transition Diagram of Planning Agent

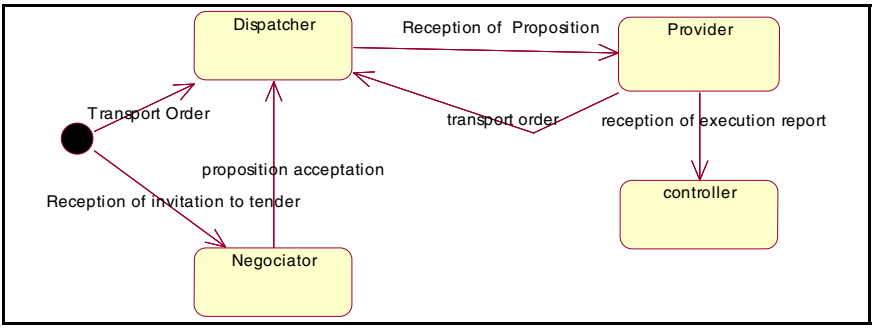


Fig. 4. State/Transition Diagram of supervisory Agent

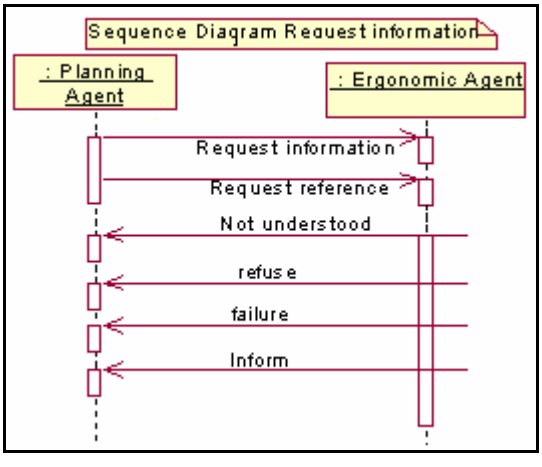


Fig. 5. Sequence diagram Request information

4 Communication and Negotiation

4.1 Communication among Agents of an MASMRT

The Agents in the MSAMRT communicate by exchanging messages. These messages express information which a transmitting agent wishes that the other agents take into account. In our approach we propose the use of the FIPA agent Communication language (FIPA TC 2002), which is concisely described below.

(bid
: sender (agent-identifier :name agentSuperviseur1@SGTR.192.176.68.2)
: receiver (set (agent-identifier :name agentPlanificateur2@SGTR.152.255.100.5))
: content "((announce new sub order Γ(a,d,l) with
l= {(l1, q1), (l2, q2),…, (ln,qn)}))"
: language fipa-sl

: ontology auction
: protocol Contract Net
)

This message is sent by the supervisory agent to all its planner agents to announce them a new elementary order Γ(a,d,l).

Figure 5 shows the sequence diagram which illustrates the interaction between the planning agent and the ergonomic agent

4.2 Negotiation

In our approach we propose the use of the Contract-Net protocol (Ouelhadj and Al, 2003) to manage the negotiation among the Supervisory Agent and its Planner Agents. It is a negotiation mechanism between two kinds of agents: contractor and manager. It is a mode of tasks allowances which functions according to the broadcast or multicast principle. It makes it possible to a manager following some exchanges with a group of agents to retain the services of an agent called contractor for the execution of a task (contract). This protocol is also described as type "mutual selection" for the signing of a contract, the selected agent must be committed towards the manager for the execution of the task, and the manager selects only the agent having provided the most advantageous proposal. The original version of the protocol described in what follows comprises three principal stages: the invitation to tender, the tender of the proposals and the attribution of contract (cf figure 6).

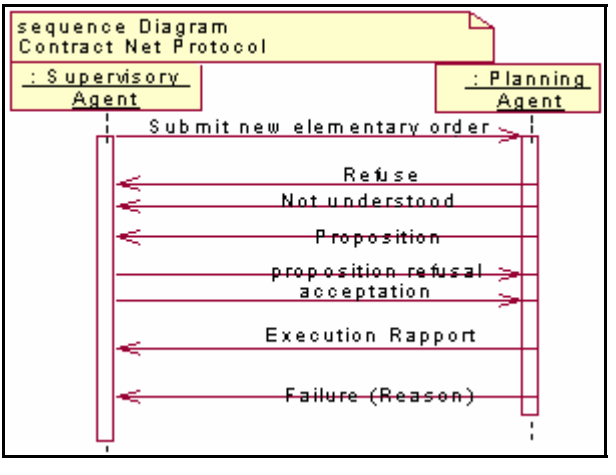


Fig. 6. Sequence diagram Contract Net Protocol

5 Tasks Planning

The tasks planning problem in transport consists in calculating the optimal roads to distribute transport orders through a set of customers. It is a hard combinatorial optimization problem. The resolution of such hard problems using traditional operational research techniques requires a prohibitory calculating time. We dealt with this

problem in a previous work, for more details see [12] and [13]. Nevertheless we give here its basic idea: we have proposed two planning strategies. The first is a static planning based on the Hopfield Neural Networks (HNN) which deals with the initial plans generation of each vehicle. In this level, we propose a co-operative planning between the various companies in order to optimize the plans of each company [12]. The second is a dynamic planning based on multiple heuristics (genetic Algorithms, Lin - Kehrnignan, Tabu search), while each one of them is activated in the suitable case [13].

6 Applications

In a previous work (Nejeoui and Al, 2006) we treated the dynamic aspect of the MASMSRT, In order to show the impact of the co-operation inter companies on the static planning, we are interested only in the latter, by application of the HNN, and we insist more particularly on the co-operation inter company via the supervisory agents.

6.1 Modeling

With an aim of illustrating our proposal, we will consider an example of a transport company which cooperates with another transport company. Each company is characterized by an identifier, its co-ordinates and its vehicles fleet. In our approach, we associate a supervisory agent to each company. We will have SA_1 and SA_2 characterized by:

Agent_Identifier	SA_1	SA_2
Company_ Name	Company1	Company2
Company_Adress	Ad_1= (-14,77)	Ad_2= (0,0)
Fleet_Seize	10	10

Each company has a homogeneous vehicles' fleet of 10 vehicles with a maximum capacity Q=1679.

Let us suppose that the supervisory agents receive the following transport orders: $\Gamma_1(a_1,d_1,l)$, $\Gamma_2(a_2,d_2,l)$, $\Gamma_3(a_3,d_3,l)$ et $\Gamma_4(a_4,d_4,l)$ distributed as follows:

Γ_1 and Γ_2 are granted to SA_1, Γ_3 and Γ_4 are granted to SA_2

In the following we show the characteristics of each Transport order:

d_i : quantity of goods requested by the customer i, L_i : its Number, (X_i,Y_i) : its position.

$\Gamma_1(a_1,d_1,l)$ a1=3321 d1=(-14,77)	**$\Gamma_2(a_2,d_2,l)$** a2=4865 d2=(-14,-7)	
Li Xi Yi di	Li Xi Yi di	16 -87 -61 718
1 -2 7 74	1 -23 -102 44	17 0 -3 4
2 -7 17 29	2 -12 11 262	18 5 -5 23
3 -2 3 9	3 -26 15 54	19 -13 -79 764
4 2 14 391	4 10 -70 9	20 -35 -8 34
5 -1 28 25	5 -33 23 5	21 10 -16 93
6 -14 24 286	6 -30 15 225	22 -1 -15 3
7 -10 31 18	7 11 -104 676	23 -19 -96 17

8 -14 77 822	8 8 -95 22	24 -17 -76 21
9 -11 80 531	9 -12 -101 120	25 -3 -8 4
10 -13 80 181	10 -6 -12 11	26 -13 9 5
11 -21 74 3	11 -29 7 815	27 -33 20 68
12 -8 12 16	12 6 -113 152	28 -53 -32 13
13 -1 7 4	13 1 -3 8	29 -7 -11 398
14 -16 83 884	14 -5 -1 183	30 -60 -32 52
15 0 3 48	15 -78 -2 6	31 4 -5 56

$\Gamma_3(a_3,d_3,l)$
a_3=1677
d_3=(-14,77)
Li Xi Yi di
1 -8 91 45
2 -23 74 4
3 -6 88 54
4 -8 88 10
5 -25 78 140
6 -5 88 58
7 -20 80 185
8 -25 89 285
9 -24 79 59
10 -5 87 11
11 -6 91 159
12 -21 90 15
13 -26 88 637
14 -25 74 4
15 -5 82 11

$\Gamma_4(a_4,d_4,l)$
a_4=5043
d_4=(0,0)
Li Xi Yi di
1 -3 0 124
2 4 1 81
3 3 -1 841
4 7 2 7
5 15 -2 406
6 10 2 37
7 -4 1 4
8 -2 -5 1066
9 3 0 687
10 -20 -2 588
11 -26 0 480
12 7 -3 175
13 -36 2 518
14 -2 -4 29

6.2 Generation of the Initial P lans

The first step consists in generating the initial plans by subdividing each transport order Γ_1, Γ_2, Γ_3 and Γ_4 in a set of elementary orders. For that, each agent subdivides its transport orders using Hopfield Neural Networks algorithm. After this stage we obtain the following structuring of plans:

SA_1					SA_2			
Orders					Orders			
Γ_1		Γ_2			Γ_3	Γ_4		
Structuring Orders		Structuring Orders			Structuring Orders	Structuring Orders		
Γ^1_1	Γ^2_1	Γ^1_2	Γ^2_2	Γ^3_2	Γ_3	Γ_{14}	Γ_{24}	Γ_{34}

Customers of each elementary order								
2,3, 7, 9,10, 11, 13, 14	1, 4, 5, 6, 8,12, 15	2, 3, 5, 6, 11, 14, 26, 27	15, 16, 17, 19, 20, 22, 28, 30	1, 4,7, 8, 9,10, 12,13, 18,21, 23,24, 25,29, 31	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	7 ,10, 1 1,1 3		2, 3,4, 5, 6, 12

After this step, each supervisory agent will have many elementary orders to negotiate and attribute to its planner agents.

6.3 Negotiation and Attribution of Plans with Co-operation between AS1 and AS2

Each supervisory agent negotiates its elementary orders with its agents while including the other supervisory agent in the negotiation process.

	Planner Agents		Elementary orders	Rounds	Cost
AS_1	Caste1 -1	^{1}APl_1	Γ^1_1	14.11.7.2.3.13.9.10	256.02
		^{1}APl_2	Γ^2_1	8.5.4.15.1.12.6	237.76
AS_2	Caste1 -2	^{1}APl_3	Γ_3	15.10.6.3.4.11.1.12.8.1 3.7.9.5.14.2	230.00
	Caste2 -1	^{2}APl_1	Γ^1_4	7.13.11.10.	78.14
		^{2}APl_2	Γ^2_4	9	11.38
		^{2}APl_3	Γ^3_4	3.12.5.6.4.2	37.77
	Cas te2-2	^{2}APl_4	Γ^1_2	14.11.6.27.5.3.26.2	117.28
		^{2}APl_5	Γ^2_2	20.15.28.30.16.9.22.17	327.46
		^{2}APl_6	Γ^3_2	25.29.10.24.23.1.9.12.7 .8.4.21.18.31.13	224.06

The figure 7 shows the solution to distribute the 4 transport orders while taking into account the co-operation between the two SA.

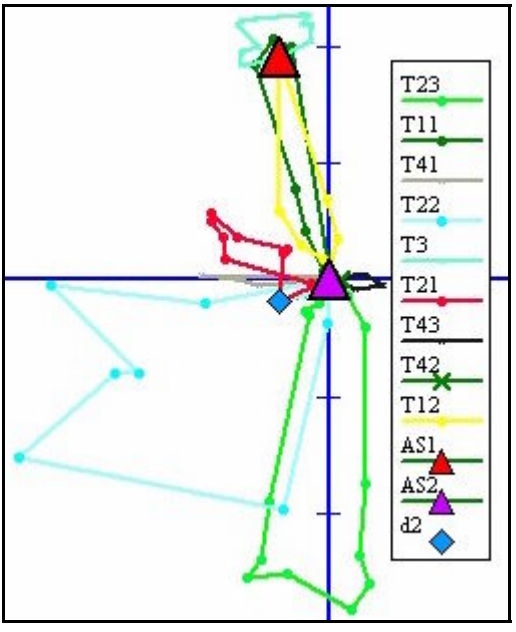


Fig. 7. Orders Distribution with cooperation

The parameters which influence the quality of the solution are the total distance traversed by the whole the PLAs to serve all ACs Requests as well as the number of PLAs necessary to achieve the transport mission (this parameter represents a significant criterion from an economic standpoint). Figure 7 shows the results which illustrate the performances of the solution obtained by the MASMRT after the integration of the co-operation techniques between companies. The results show that the elementary order Γ^1_2, Γ^2_2, Γ_3.

We remark that the introduction of the co-operation techniques in the planning process of the companies has enabled the MASMRT to pass from a total cost of 2221.7981 to serve the 4 transport orders without taking into account of the co-operation between AS1 and AS2 for a cost of 1505.3596 when taking into account the co-operation, which means that the system has saved about 716.4385 i.e. 32.24% of the total cost.

This large gain of performance can easily justify the investment necessary to integrate a co-operative planning into the decision process of the road transport companies.

7 Conclusion and Perspectives

In this paper, we have presented an approach based on the classification of agents for modeling the MSRT. The principal reason that makes the MSRT especially suitable for using techniques from Distributed AI is the complexity of the real-time scheduling of transport orders which comes in an asynchronous and dynamic way.

The suggested approach is very promising in the field of design and analysis of distributed systems characterized by an inherent distribution of knowledge and control; such is the case in all means of transport: Road, maritime, railway and air. In a future work we project to adopt this approach to model and design a Multi-Agents System of Management Intermodal Transport which includes all other means of transport.

References

1. Crainic, T.G.: Long haul freight transportation. In: Hall, R.W. (ed.) Handbook of Transportation Science, 2nd edn. Kluwer, Dordrecht (2002)
2. Regan, A., Mahmassani, H.S., Jaillet, P.: Evaluation of dynamic fleet management systems simulation framework. Transportation Research Record 1648, 176–184 (1998)
3. Jensen, L.K., Demazeau, Y., Kristensen, B.B.: Tailoring an agent architecture to a flexible platform suitable for cooperative robotics. In: Mařík, V., Müller, J.P., Pěchouček, M. (eds.) CEEMAS 2003. LNCS, vol. 2691, pp. 363–372. Springer, Heidelberg (2003)
4. Ferber, J.: «Les systèmes multi-agents. Vers une intelligence collective», InterEditions, Paris (1995)
5. Elfazziki, A.: Une approche orientée agent pour l'analyse et la conception des systèmes de contrôle des processus industriels.-2002. Thèse de doctorat d'état en informatique. Faculté des sciences Semlalia Marrakech (Septembre 2002)
6. Smith, R.G.: The contract net protocol: high-level communication and control in a distributed problem solver. IEEE Trans. on Computers 29(12) (1980)
7. Muller, P.A.: Modélisation objet avec UML, édition Eyrolles (2000)
8. Bauer, B.: UML class diagrams revisited in the context of agent-based systems. In: Wooldridge, M.J., Weiß, G., Ciancarini, P. (eds.) AOSE 2001. LNCS, vol. 2222, p. 101. Springer, Heidelberg (2002)
9. `http://www.FIPA.org`
10. Louis, V., Martinez, T.: An operational model for the FIPA-ACL semantics. In: Agent Communication Workshop, AAMAS 2005 (2005)
11. Ouelhadj, D., Cowling, P.I., Petrovic, S.: Contract Net Protocol for Cooperative Optimisation and Dynamic Robust Scheduling of Steel Production. In: Intelligent Systems Design and Applications (ISDA 2003), Tulsa, OK, USA (2003)
12. Nejeoui, A., Elfazziki, A., Sadgal, M., Aitouahman, A.: A Hopfield-Neural Network to Deal with Tasks Planning in a Multi-Agent System of Transport. Wseas Transactions On Systems 5(1), 180–187 (2006)
13. Elfazziki, A., Nejeoui, A., Sadgal, M.: Une approche multi-agents pour la modélisation et l'optimisation des Systèmes de gestion de transport maritime. RIST revue d'information scientifique et technique (2006)

An Agent-Based Organizational Model for Cooperative Information Gathering

Issam Bouslimi[1], Khaled Ghédira[1], and Chihab Hanachi[2]

[1] Institut Supérieur de Gestion de Tunis & ENSI, SOIE Laboratory
41, Rue de la Liberté, Cité Bouchoucha , le Bardo 2000, Tunisie
bouslimi.issam@isg.rnu.tn, khaled.ghedira@isg.rnu.tn
[2] IRIT Laboratory,
University Toulouse 1, 31042 Toulouse Cedex, France
hanachi@univ-tlse1.fr

Abstract. Organization modeling is recognized as an essential mechanism for structuring the design of Multi-Agent Systems and coordinating their executions. In this paper, an organizational model for Cooperative Information Gathering Systems (CIGS) is proposed. This model has three levels of abstraction, starting with a general system description and progressively transformed onto a concrete organization. This model is organized around reusable and conceptual basic bricks corresponding to roles and protocols. The roles and their participations in the protocols are defined according to a deontic structure inspired from the GAIA methodology, but specified formally using Petri Nets with Objects. The originalities and the advantages of our proposal are i) the definition of an organizational model adapted to the CIG ii) a formalization of the roles, which makes it possible to analyze and simulate them before their deployment iii) the possibility to reuse the components of the model.

Keywords: cooperative information gathering, agent, organizational model, Petri nets.

1 Introduction

Cooperative Information Gathering (CIG) is an extension of conventional Information Retrieval with the following three additive aspects:

– Information Sources (IS) are distributed, autonomous, heterogeneous and possibly redundant [11].
– A CIG task is not a simple database query but rather a problem that has to be decomposed into several interdependent sub-problems [3], [9].
– The answer to a question requires a co-operation between several information retrieval systems [7] and must involve and interact intelligently with the user. Information retrieval becomes a distributed problem solving activity [10] where a group of information retrieval agents is dynamically selected and coordinated to solve a given problem, jointly and in co-operation with the user.

CIG raises numerous specific problems that can be summarized in three questions - What? Who? and How? - Corresponding respectively to the following aspects:

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 180–189, 2009.

– *the informational aspect,* which determines the universe of discourse: This aspect is related to the modelling of the information contained in the Information Sources but also to the view provided to each user on this information. This question is difficult to solve, given that the Information Sources to be explored are unknown a priori, possibly heterogeneous and autonomous.
– *the organizational aspect,* which identifies the actors involved in the system, the links between them, and the protocols that rule their interactions. At this level, it is necessary to take into account the user but also original components born from the development of this technology such as the Translators (Wrappers), Brokers, Mediators or Facilitators. The challenge is then to determine the most adequate organizational structure to organize this system.
– *the task aspect*, which specifies the information gathering process: It requires the definition of a task description language which takes into account a multitude of phenomena such as distribution of Information Sources, openness of the universe, parallel execution of agents but also their cooperation.

The study of many systems [16] (TSIMMIS [13], RETSINA [15], MACRON [4], IRO-DB [6], MASTER-Web [7]), representative of the state of the art, shows that the organizational point of view is insufficiently treated. These systems gave place to informational models (mainly based on ontologies), or task models (mainly based on the AND/OR graph formalism), but very few of them investigated the organizational aspect like a full dimension. It results in an absence of abstractions, conceptual tools (notation, formal models) and specific methods to guide at the same time the design and the development of a Cooperative Information Gathering System (CIGS for short). It thus appears necessary to propose methods and tools to specify, validate and implement CIGS. As organizational methods such as GAIA [17], AALAADIN [5], or MOISE+ ([1]) show it, an organizational perspective is useful to guide the design and development process of MASs, facilitate their intelligibility, allow to reuse some of the components, and structure their executions.

This paper proposes an Organizational Model (OM for short), which completes the informational and task models, and that interacts with them. In order to facilitate the design and development process of a CIGS, this OM is structured in three levels, starting with an abstract description of the system, which will be refined progressively while going down from one level to another. It describes i) the roles, or functions, which must be ensured by a CIGS, and ii) the protocols according to which these roles interact. The roles are expressed in accordance with the structure suggested by the GAIA methodology. However, to represent the concurrent characteristics of these roles, and their deontic aspects, we propose to model their behavior (rights and duties) using Petri Nets with Objects (PNO). This original formalization of the roles has three advantages. It allows the designer to: i) represent all the deontic patterns appearing in GAIA, ii) refer to the informational and task models within a uniform framework, and finally iii) analyze and simulate the roles before their deployment. This model also integrates specific components of the CIG field such as the wrappers, matchmaker and mediator.

The remainder of the paper is organized as follows. Section 2 gives an overview of our organizational model by clarifying the three levels and their interactions. We develop in section 3 the first level called the "Role Model" by clarifying the “typical” roles and their interaction protocol. Section 4 concludes this work.

2 The Organization Model

2.1 Some Preliminary Definitions

In the CIG field, *a task occurrence* is a question submitted by a user (for example, the organization of a trip to Tunis City in December 2006). The solving process for answering a question is not unspecified but obeys to a predefined schema called a *task*. A task does not correspond to a single question (problem) but to a class of *questions*, which differ only by the value of their parameters. Thus, "the organization of a trip in a given *city* during a given *period* of the year 2006" corresponds to a class of questions which admits two parameters "the city" and "the period".

2.2 The Three Levels of the Organizational Model

An OM deals with the organization of a system i.e. the definition of its components through their role and their behavior. To define an OM, it is beforehand necessary to specify the functions, which must be fulfilled by the system being modeled. The analysis of these functions will make it possible to identify the roles responsible for these functions, and the links, which exist between these roles.

The designer must also identify and represent recurring patterns to be reused. Two types of patterns are to be considered: those relating to the behaviors of the agents, and those relating to the interactions between agents. Two key concepts make it possible to represent such patterns: respectively the concept of role and the concept of interaction protocol [8]. *A role* constitutes an abstraction of the behavior of the agents, and an interaction protocol defines the structure of the interactions between agents [2] through the roles they play. These two abstractions are interesting because they make it possible to design the conceptual model of an organization without referring to the concrete agents likely to take part in the organization, but rather to the qualities they must feature. This conceptual model, once established, can be analyzed and simulated before being instantiated to be executed. At run time, occurrences of the conceptual model (case) are initialized and carried out by distributing the roles to the suitable agents. An agent is then likely to play several roles, and a role can be played by several agents.

The recurrence of behavioral and interaction patterns can intervene at two levels. Some patterns are repeated only within one same task (i.e. they intervene for each occurrence of the task), whereas others intervene systematically in all the occurrences of all the tasks. This observation leads us to define classes of roles, in the same way as [12], called Typical Roles. A *Typical Role* represents a class of roles independent of a task of particular CIGS, and having the same types of actions and interactions. A Typical Role is a high level class, invariant and having all the advantages of a class: it can be specialized or instantiated.

It results from this, an Organizational Model in three levels of abstraction:

1. On the first level, a Role Model (RM) is defined by a set of Typical Roles and the Interactions Types which exist between them. This level of description is the most abstract; it is independent of the task.
2. At the second level we find Organizational Structures (OS). An OS is a specialization of the Role Model which defines the structure of an organization specific to a

task. It is defined by a set of specialized roles and the interactions between these roles.
3. At the concrete level we find Concrete Organizations (or more simply an Organization). An Organization is an instance of an Organizational Structure where agents play roles and interact for solving in cooperation a task instance.

In the remainder of this paper, for sake of place, we will only detail the Role Model.

3 The Role Model

3.1 The Typical Roles and Their Interaction Protocols

Thanks to a functional analysis, we have identified the typical roles, which necessarily intervene in any CIGS. In fact, we distinguish two categories of agents. The first category corresponds to agents directly related to the definition and the execution of the tasks, and they are instantiated and deployed for each occurrence of a task:

– **The Mediator** supervises the execution of each task, reformulates and decomposes the query of the user, and composes the final result.
– **Coordinators** are agents in charge of elementary tasks. They communicate with the matchmaker to find external information agents (translators) able to retrieve information, wait for that information, and communicate it to the mediator and to other coordinators if needed via the Information Exchange Manager. They can iterate this process in case of failure.
– **Information Exchange Manager** (IEM) supervises the exchange of information between coordinators.
The second category of performers pre-exists to the task modeling activity and constitutes resources exploited by the CIG process:
– **The Matchmaker** provides references towards external informational agents able to carry out elementary tasks.
– **User** provides the initial query, interacts with the Mediator to guide its execution and receive the response to its initial query.
– **Translators** (translators or wrappers) are external agents recruited via the matchmaker and in charge of retrieving information. A translator is associated with an Information Source (IS); it translates a coordinator question into the query language of its source and converts results extracted from its source to a format understandable by the coordinator.

To define the structure of a typical role, we took as a starting point the type structure of the GAIA methodology, and made some adaptations.

A typical role is a class of roles defined by Duties and Rights. *The Duties* are defined by a set of abstract Actions, a set of *Interventions*, their coordination rules, and the Invariants. *The abstract Actions* are the internal actions that the typical role can perform without interacting with other roles. In our proposal, to preserve the autonomy of the agents, which play a role, the local actions are represented in an abstract way by giving their name but without referring to their implementation. *The Interventions* correspond to the types of messages in which a typical role is implied as a transmitter or a receiver.

The coordination rules represent the authorized sequences of actions and interventions. *The Invariants* represent a predicate that must be verified by the different possible states of a typical role. *The Rights* are defined by Resources and Authorizations. *The Resources* correspond to the set of resources accessible by the typical role while executing its actions. A resource can be the informational model, the task model or the knowledge base of a typical role. *The Authorizations* define the type of access a typical role can have on its resources. The different types of access are reading, writing, modifying or consuming.

It should be noted that the protocols are not defined as a distinct (separate) objects but are distributed in each standard role. In the remainder, we will use the term *protocol* to indicate a global (entire) protocol, and the expression *contribution to a protocol* to indicate the local part of a role in a protocol: it is the projection of a protocol according to a role. Given a protocol P and a typical role R, the contribution of R to P is the set of interventions of R in P with their coordination constraints.

We give in section 5 a Petri Net with Objects formalization of the concept of typical role.

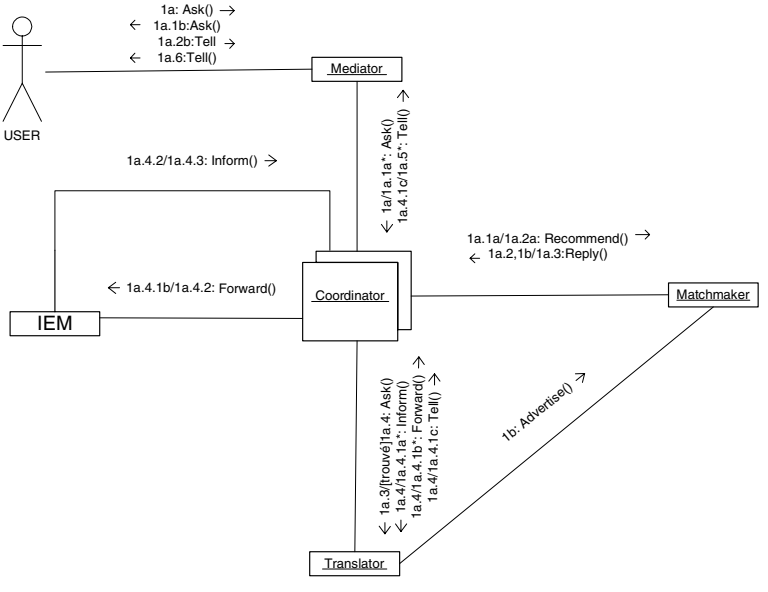


Fig. 1. Collaboration diagram between typical roles

3.2 Global View on the Interactions between Typical Roles

Before detailing the typical roles in accordance with the structure defined in section 3.1, the collaboration diagram given in figure 1 represents a global view of typical roles, their interventions and the order of these interventions. Considering that the interventions operating under the causal dependency relation form a protocol, it is possible to extract from this diagram the two following interaction protocols:

– The answering process to a question, involving all the typical roles of our model. It begins with the user question (1a) and ends with the final answer which is returned to him (1a.6).
– The publication of a capacity, involving a Translator and a Matchmaker, and limited to an exchange (1b).

3.3 Specifying Typical Roles Using the Petri Nets with Objects Formalism

As we have previously precise it, the definition of a typical role includes its actions, its interventions, the invariants, the coordination rules of its various activities and interventions, its resources and the access authorizations to each resource. To represent these characteristics and the concurrent activities of the agents intervening in a CIG process, we will use Petri nets with Object (PNO) [14] known to be an adequate formalism for specifying, simulating and verifying parallel and distributed systems in a formal way.

3.3.1 Overview of the Petri Nets with Objects Formalism

Petri Nets with Objects (PNO) [14] are a formalism combining coherently Petri nets (PN) technology and the Object-Oriented (OO) approach. While PN are very suitable to express the dynamic and parallel behavior of a system, the OO approach permits the modeling and the structuring of its active (actor) and passive (information) entities. In a conventional PN, tokens are atomic, whereas they are objects in a PNO. As any PN, a PNO is made up of places, arcs and transitions, but in a PNO, they are labeled with inscriptions referring to the handled objects. More precisely, a PNO features the following additional characteristics:

– Places are typed. The type of a place is a (list of) type of an (list of) object(s). A token is a value matching the type of a place such as a (list of) constant (e.g. 2 or 'hello'), an instance of an object class, or a reference towards such an instance. The value of a place is a set of tokens it contains.
– Arcs are labelled with parameters. Each arc is labelled with a (list of) variable of the same type, as the place the arc is connected to. The variables on the arcs surrounding a transition serve as formal parameters of that transition and define the flow of tokens from input to output places. Arcs from places to a transition determine the possible condition of the transition: a transition may occur (or is possible) if there exists a binding of its input variables with tokens lying in its input places.
– Each transition is a complex structure made up of three components: a precondition, an action and emission rules. A transition may be guarded by a precondition, i.e. a side-effect free Boolean expression involving input variables. In this case, the transition is only permitted by a binding if this binding evaluates the precondition to be true. Passing a transition through depends on the precondition, on the location of tokens and also on their value. Most transitions also include an action, which consists in a piece of code in which transitions' variables may appear and object methods be invoked. This action is executed at each occurrence of the transition and it processes the values of tokens. Finally, a transition may include a set of emission rules i.e. side-effect free Boolean expressions that determine the output arcs that are actually activated after the execution of the action.

In order to allow the hierarchical design of nets and to account for the communications between nets, it is possible to distinguish entry places and result places. *Entry places* are intended to receive tokens from the environment and have only output transitions, while conversely *result places* are intended to supply tokens to the environment and have only input transitions. Then, several nets may be merged into a single one, by merging places: one result place of a sub-net is merged with an entry place of another sub-net, or several couples of places are one-to-one merged.

3.3.2 Typical Roles Representation with PNO

We represent the behaviour of a typical role by a PNO by following the six following principles (illustrated with the example of the mediator of figure 2):

– the typical roles with which it interacts are delimited by rectangles where only communication places appear (called interface). A place of interface between two typical roles is an input place for one and an output place for the other.
– the internal actions carried out by a role are represented by actions associated with the transitions,
– the interventions are represented by arcs labelled by KQML performatives and which connect a transition from the transmitting typical role to a place of the receiver typical role,
– An invariant is a property, which is true for any accessible marking starting from an initial marking. There exist numerous Petri Nets analysis techniques to deduce them.

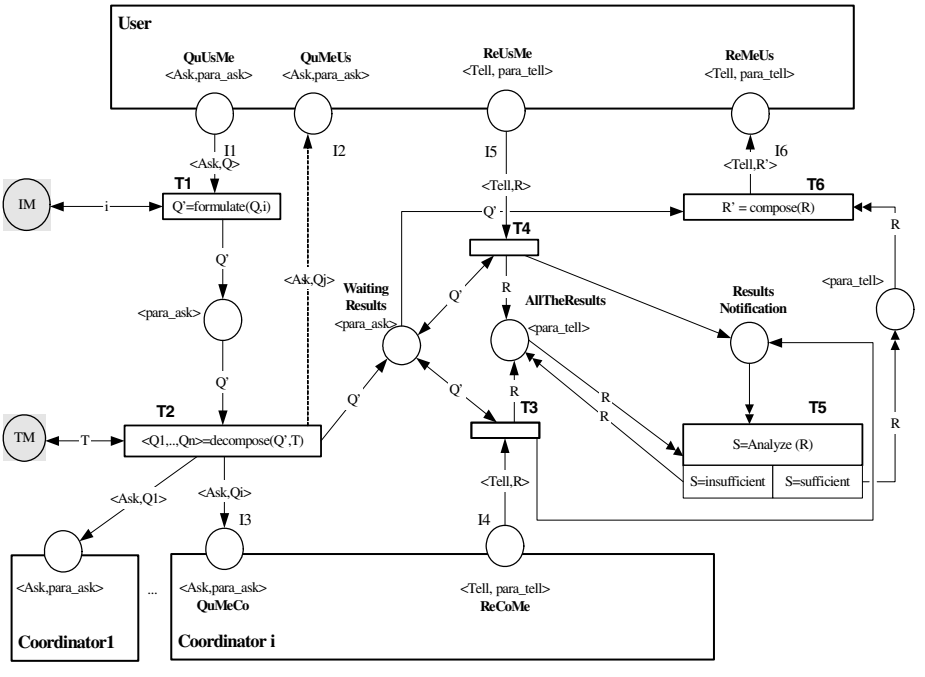


Fig. 2. Formal Specification of the typical role Mediator with PNO

– The resources are represented by grey places to distinguish them from the other places,
– The authorizations are represented by arcs, which connect transitions to resources.

The typical role Mediator, which behavior is shown in Figure 2, interacts with the Typical Roles *User* and *Coordinator*. It has four interactions with the User role corresponding to the four interface places on the top of the figure, and two interface places with one or several Coordinators.The Ressources used by the Mediator are the Informational Model (*IM*) and the Task Model (*TM*). The initial state of the mediator is when the *QuUsMe* place (for *QuestionUserMediator*) contains a question in the form of performative *Ask* with its parameters. Its final state is when the *AnMeUs* place contains the final answer transmitted using a performative *tell*.

The behaviour of the *Mediator* can be broken up into three stages, identifiable in the network:

1. The left part of the net made up of transitions *T1* and *T2*, allows the Mediator to accept the question of a user, to reformulate it, to decompose it into sub-questions, and to submit them to the coordinators and/or to the user. After the reception of the question of the user, the Mediator reformulates the question (Transition T1) taking into account the context of the question and the preferences of the user, which appear in the *IM*. After the reformulation of the question, the *Mediator* decomposes the question (Transition T2) into sub-questions submitted to the *Coordinators* and possibly to the *User*. To represent the fact that this interaction is optional, we use a dotted arc. Then, the *Mediator* waits for the answers of these sub-questions (*AwaitingResults* place)
2. The central part of the net, composed of transitions *T3* and *T4*, allows the *Mediator* to collect the answers to the sub-questions. When the *Mediator* receives a result from *a Coordinator* or from *a user* it stores this result (respectively using *T3* or *T4*) in the *AllTheResults* place, and announces the presence of new results by putting a token in the *ResultsNotification* place.
3. The right part of the network, made of the transitions *T5* and *T6*, analyzes the set of the stored results (*T5*), which, if they are considered to be sufficient, are merged (*T6*) to obtain the final result and submit this result to the user. If the results are insufficient, the *Mediator* goes on waiting complementary results. It will start again a new analysis (*T5*) when new results arrive. The presence of one or more tokens in the *ResultsNotification* place informs the *Mediator* of this event.

The language of the net, which gives all the possible behaviours, is $T1.T2.((T3|T4).T5)^n.T6$ where n is the number of results received by the mediator in response to the questions submitted to the user and to the coordinators. Among the *Invariants* of this specification, we can state that when the mediator reaches its final state (*ReMeUs* place), it is not any more waiting for results since the *T6* transition which puts it in the final state consumes all the tokens which are in the *WaitingResults* place.

4 Conclusion

This paper has proposed an Organizational Model for supporting the design and the execution of a Cooperative Information Gathering System. This organization, with three progressive levels of abstraction, has the following software engineering advantages:

– *Validation and simulation* are possible thanks to the Petri Net with Objects formalism used to represent the typical role specifications.
– *Reusabillity* of conceptual components such as typical roles and their interventions in the protocols.
– *Self-organization* since a CIGS adapts the number and the nature of the agents according to the question and the progress of the answering process. An original concrete organization is established for each question, and it evolves according to the solving process.

Although organization modeling has become increasingly important in MAS, there have been very few real world applications following this approach and no tentative in the CIG area. Therefore, applying this approach to CIG is another innovative aspect of this paper. This work can also be considered as a contribution to the agent- oriented methodologies field (see GAIA, AALAADIN or MOISE+) where the concepts of organization, roles and protocols are fundamental. Our work formalizes the concept of role and provides possibilities of simulation and validation inherited from Petri Nets. This work can also be extended by considering all the roles as Web services, allowing roles to be published, discovered, invoked and combined. This direction requires translating the PNO formalism into a semantic web language like OWL-S for example.

References

1. Hübner, J.F., Sichman, J.S., Boissier, O.: Using the Moise+ for a Cooperative Framework of MAS Reorganisation. In: Bazzan, A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS, vol. 3171, pp. 506–515. Springer, Heidelberg (2004)
2. Cranefield, S., Purvis, M., Nowostawski, M., Hwang, P.: Ontologies for Interaction Protocols. In: Falcone, R., Barber, S., Korba, L., Singh, M.P. (eds.) AAMAS 2002. LNCS, vol. 2631. Springer, Heidelberg (2003)
3. Decker, K., Pannu, A., Sycara, K., Williamson, M.: Designing behaviors for information agents. In: Proceedings of the 1st ntl. Conf. On Autonomous Agents, Marina del Rey, Février, pp. 404–413 (1997)
4. Decker, K., Lesser, V.: MACRON: An Architecture for Multi-Agent Cooperative Information Gathering. In: Prasad, N., Wagner, T. (eds.), `ftp://ftp.cs.umass.edu/pub/techrept/techreport/1995/UM-CS- 1995-099.ps`
5. Ferber, J., Gutknecht, O.: A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems. In: Proceedings of the 3rd International Conference on Multi-Agents Systems (ICMAS). IEEE CS Press, Los Alamitos (1998)
6. Gardarin, G.: Multimedia Federated Databases on Intranets: Web-Enabling IRO-DB. In: Tjoa, A.M. (ed.) DEXA 1997. LNCS, vol. 1308. Springer, Heidelberg (1997)
7. Freitas, F.L.G., Bittencourt, G.: An Ontology-based Architecture for Cooperative Information Agents. In: IJCAI 2003, pp. 37–42 (2003)
8. Hanachi, C., Sibertin-Blanc, C.: Protocol Moderators as Active Middle-Agents in Multi-Agent Systems. Journal of Autonomous Agents and Multiagent Systems (JAAMAS) 8(2) (2004)
9. Hanachi, C., Sibertin-Blanc, C., Tout, H.: A Task Model for Cooperative Information Gathering. In: IEEE International Conference on Systems, Man and Cybernitics, Hammamet, Tunisia (October 2002)

10. Oates, T., Prasad, M.V.N., Lesser, V.: Cooperative Information Gathering: A Distributed Problem Solving Approach. UMass Computer Science Technical Report 94-66-version 2 (1994)
11. Ouksel, A.M.: Semantic Interoperability in Global Information Systems: A brief introduction to the research area and the special section", Special section on Semantic Interoperability in Global Information Systems. In: Ouksel, A.M., Sheth, A. (eds.) SIGMOD RECORD, vol. 28(1) (1999)
12. Ould, M.A.: Business Processes Modelling and Analysis for Re-engineering And Improvement. John Wiley & Sons, Chichester (1995)
13. Papakonstantinou, Y., Garcia-Molina, H., Ullman, J.: Medmaker: A mediation system based on declarative specifications, `ftp://db.stanford.edu/pub/papakonstantinou/1995/medmaker.ps`
14. Sibertin-Blanc, C.: High-level Petri nets with Data structure. In: 6th European workshop on Petri nets and applications, Espoo (Finland), Juin (1985)
15. Sycara, K., Pannu, A.S.: The RETSINA multiagent system: Towards integrating planning, execution and information gathering. In: Sycara, K., Wooldridge, M.J. (eds.) Proceedings of the 2nd International Conference on Autonomous Agents (Agents 1998), New York, pp. 350–351 (1998)
16. Tout, H.: Engineering Cooperative Information Gathering Systems, PHD thesis University Toulouse 1 (September 2003)
17. Zambonelli, F., Jennings, N.R., Wooldridge, M.: Developing Multiagent Systems: The Gaia Methodology. ACM Transactions on Software Engineering Methodology 12(3), 317–370 (2003)

A Dynamic Grid Scheduler with a Resource Selection Policy

Said Elnaffar[1] and Nguyen The Loc[2]

[1] College of IT, UAE University, UAE
elnaffar@uaeu.ac.ae
[2] Hanoi National University of Eduation, VietNam
locnt@hnue.edu.vn

Abstract. Many algorithms in the literature have been targeting the problem of scheduling divisible workloads (those loads that are amenable to partitioning in any number of chunks). Unfortunately, such algorithms have a number of shortcomings such as the sole reliance in their computations on CPU speed, and the assumption that a definite set of workers are available and must participate in processing the load. These constraints limit the utility of such algorithms and make them impractical for a computing platform such as the Grid. In this paper, we propose an algorithm, MRRS, that overcomes these limitations and adopts a worker selection policy that aims at minimizing the execution time. The MRRS has been evaluated against other scheduling algorithms such as UMR and LP and showed better results.

Keywords: Divisible load, task scheduling, resource selection.

1 Introduction

By definition, a divisible load is "a load that can be partitioned into any arbitrary number of load fractions" [1]. This kind of workload arises in many domains of Grid computing [2] such as protein sequence analysis and the simulation of cellular micro physiology. Per the Divisible Load Theory [1], the scheduling problem is identified as: "Given an arbitrary divisible workload, in what proportion should the workload be partitioned and distributed among the workers so that the entire workload is processed in the shortest possible time."

The first multi-round (MI) algorithm, introduced by Bharadwaj [1], utilizes the overlapping between communication and computation processes at workers. In the MI algorithm the number of rounds is predefined and fixed. It overlooks communication and computation latencies. The studies in [3] focus on Affine model in which computation and communication latencies are different from zero. Yang et al. [4] through their UMR (Uniform Multi-Round) algorithm, designed a better algorithm that extends the MI by considering latencies. However, in the UMR, the size of workload chunks delivered to workers is solely calculated based on worker's CPU power; the other key system parameters, such as network bandwidth, are not factored in. Beaumont [3] proposes another multi round

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 190–200, 2009.

scheduling algorithm that fixes the execution time for each round. This enabled the author to give analytical proof of the algorithm's asymptotic optimality. But the influence of this assumption on the utilization of transfer-execution overlap is questionable.

One apparent shortcoming in many scheduling algorithms [1,3,4] is the abandon of a selection policy for the best subset of available workers. The selection of the best workers is a chief task especially in a large platform such as the Grid that agglomerates hundreds or thousands of workers. The present scheduling algorithms rely on the assumption that all workers are available and must participate in processing load partitions. This assumption is not always realistic and may not lead to the minimum execution time (henceforth it is called makespan).

In this paper, we propose a new scheduling algorithm, MRRS (Multi-round Scheduling with Resource Selection), which is inspired by the UMR algorithm. MRRS is superior to UMR with respect to two aspects. First, unlike the UMR, which relies primarily in its computation on the CPU speed, MRRS factors in several other parameters such as bandwidth capacity and all types of latencies (CPU and Bandwidth) which renders the MRRS a more realistic model. Second, the UMR assumes that all workers are available and should participate in the workload processing, which is not practical especially in a computing environment such as the Grid. The MRRS, on the other hand, is equipped with a worker selection policy that works on selecting the best workers that can produce shorter makespan. As a result, our experiments show that our MRRS algorithm outperforms previously proposed algorithm including the UMR.

The rest of this paper is organized as follows. Section 2 briefly describes the heterogeneous computation platform. Section 3 and Section 4 explain the workload partitioning mechanism and the worker selection strategy, respectively, that are adopted by the MRRS algorithm. Section 5 describes the simulation experiments we have conducted in order to evaluate our work. Section 6 concludes the paper.

2 Heterogeneous Computing Platform

Let us consider a computation Grid, in that, a master process has access to N worker processes and each process runs in a particular computer. The master can divide the total workload into arbitrary chunks and delivers them to appropriate workers. We assume that the master uses its network connection in a sequential fashion, i.e., it does not send chunks to multiple workers simultaneously. The communication and computation platforms of our system are heterogeneous. Workers can receive data from network and perform computation simultaneously. The following notations will be used throughout this paper:

- W_i: worker number i.
- N: total number of available workers.
- m: the number of rounds.
- n: total number of workers that are actually selected to process the workload
- L_{total}: the total amount of workload (flop).

- $chunk_{j,i}$: the fraction of total workload L_{total} that the master deliver to worker i in round j $(i = 1, 2, ..., N, j = 1, 2, ..., M)$.
- S_i: computation speed of the W_i measured by the number of units of workload performed per second (flop/s).
- B_i: the data transfer rate of the connection link between the master and W_i (flop/s).
- $Tcomp_{j,i}$: computation time required for W_i to process $chunk_{j,i}$.
- $cLat_i$: the fixed overhead time (second) needed by Wi to start computation.
- $nLat_i$: the overhead time (second) incurred by the master to initiate a data transfer to W_i. We denote total latencies by $Lat_i = cLat_i + nLat_i$.
- $Tcomm_{j,i}$: communication time required for master to send $chunk_{j,i}$ to W_i

$$Tcomm_{j,i} = nLat_i + \frac{chunk_{j,i}}{B_i}; \quad Tcomp_{j,i} = cLat_i + \frac{chunk_{j,i}}{ES_i} \tag{1}$$

- $round_j$: the fraction of workload dispatched during round j

$$round_j = chunk_{j,1} + chunk_{j,2} + ... + chunk_{j,n} \tag{2}$$

We fix the time required for each worker to perform communication and computation during each round

$$cLat_i + \frac{chunk_{j,i}}{S_i} + nLat_i + \frac{chunk_{j,i}}{B_i} = const_j; \tag{3}$$

If we let $A_i = B_i S_i/(B_i + S_i)$ so we have

$$chunk_{j,i} = \alpha_i \times round_j + \beta_i \tag{4}$$

where

$$\alpha_i = \frac{A_i}{\sum_{i=1}^{n} A_i}; \quad \beta_i = A_i \frac{\sum_{k=1}^{n} A_k \left(Lat_k - Lat_i\right)}{\sum_{k=1}^{n} A_k} \tag{5}$$

3 Workload Partitioning

In the following subsections, we induce chunk sizes (Section 3.1) and determine the parameters of the initial scheduling round (Section 3.2). We refer the reader to [4] for more information and detailed derivations.

3.1 Induction Relation for Chunk Sizes

Figure 1 depicts the operation of our algorithm, where the computation and communication have been overlapped. At time T_1, the master starts sending $round_{(j+1)}$ amount of load to all workers and the last worker W_n starts computation $chunk_j$ concurrently. To fully utilize the network bandwidth, the dispatching of the master and the computation of W_n should finish at the same time T_2:

$$\sum_{i=1}^{n} \left(nLat_i + \frac{chunk_{j+1,i}}{B_i} \right) = \frac{chunk_{j,n}}{S_n} + cLat_n \tag{6}$$

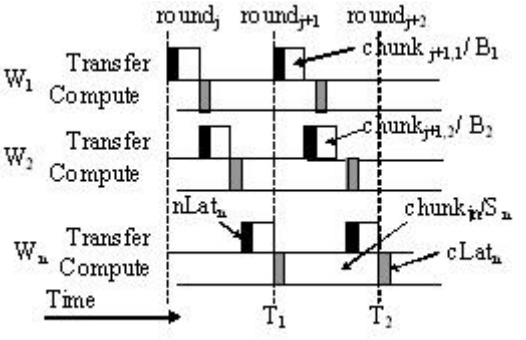


Fig. 1. Dispatching Load Chunks Using the MRRS Algorithm

If we replace $chunk_{j+1,i}$ and $chunk_{j,n}$ by their expression in (1) we derive:

$$round_{j+1} = round_j \times \theta + \mu \tag{7}$$

where

$$\theta = \frac{B_n}{(B_n + S_n)\sum_{i=1}^{n} \frac{S_i}{B_i+S_i}}; \quad \mu = \frac{\frac{\beta_n}{S_n} + cLat_n - \sum_{i=1}^{n}\left(nLat_i + \frac{\beta_i}{B_i}\right)}{\sum_{i=1}^{n} \frac{\alpha_i}{\beta_i}} \tag{8}$$

From induction equation (7) we can compute:

$$round_j = \theta^j(round_0 - \eta) + \eta \tag{9}$$

where

$$\eta = \frac{\beta_n + cLat_n - \sum_{i=1}^{n}\left(nLat_i + \frac{\beta_i}{B_i}\right)}{\sum_{i=1}^{n}\left(\frac{\alpha_i}{B_i}\right) - \frac{\alpha_n}{S_n}} \tag{10}$$

3.2 Determining the Parameters of the Initial Round

In this section we compute the optimal number of rounds, m, and the size of the initial load fragment that should be distributed to workers in the first round, $round_0$. If we let $F(m, round_0)$ denote the makespan, then from Figure 1 we can seen that

$$\begin{aligned} &F(m, round_0) = \sum_{i=1}^{n}\left(\frac{chunk_{0,i}}{B_i} + nLat_i\right) + \sum_{j=0}^{m-1}\left(\frac{chunk_{j,n}}{S_n} + cLat_n\right) = \\ &= round_0\left(\sum_{i=1}^{n}\frac{\alpha_i}{B_i} + \frac{\alpha_n(1-\theta^m)}{S_n(1-\theta)}\right) + \sum_{i=1}^{n}\left(\frac{\beta_j}{B_i} + nLat_i\right) + \\ &+m\left(cLat_n + \frac{\alpha_n\eta+\beta_n}{S_n}\right) - \frac{\alpha_n\eta(1-\theta^m)}{1-\theta} \end{aligned} \tag{11}$$

Our objective is to minimize the makespan $F(m, round_0)$, subject to:

$$\sum_{j=0}^{m-1} round_j = L_{total} \tag{12}$$

We use Lagrangian method [5] to solve this constrained minimization problem. The minimum value of function $F(m, round_0)$ can be found by solving the following equation system:

$$\left\{ \frac{\partial L}{\partial \lambda} = 0; \frac{\partial L}{\partial m} = 0; \frac{\partial L}{\partial round_0} = 0 \right\} \tag{13}$$

where:

- λ Lagrange multiplier.
- $L(m, round_0)$: Lagrangian function which is defined as: $L(m, round_0, \lambda) = F(m, round_0) + \lambda G(m, round_0)$

After solving this equation system we obtain m. Using (11) one can then compute $round_0$. At last, using (9) and (4) we will obtain the value of $round_j$ and $chunk_{j,i}$ respectively ($i = 1..n, j = 1..m$).

Algorithm 1: $Worker\_Selection(V)$
Begin
Search $W_n \in V$ such that: $\frac{B_n}{(B_n+S_n)} \leq \frac{B_i}{B_i+S_i} \quad \forall W_i \in V$
  $V_1^* = Brand\_and\_bound(V)$;
  $V_2^* = Greedy(V, \text{“}\theta < 1\text{“})$;
  $V_3^* = Greedy(V, \text{“}\theta = 1\text{“})$;
  Select $V^* \in V_1^*, V_2^*, V_3^*$ such that $m(V^*) = min(m_1(V_1^*), m_2(V_2^*), m_3(V_3^*))$;
  Return (V^*);
End

4 Worker Selection Policy

Let V denote the original set of N available workers ($|V| = N$). In this section we explain our worker selection policy that aims at finding the best subset $V^*(V^* \subseteq V, |V*| = n)$ that minimizes the makespan. If W_i denotes worker i, then W_n denotes the last worker receives load chunks in a round, and W_1 denotes the first worker that receives chunks in a round. Algorithm 1 outlines our selection algorithm. It starts with finding the last worker (W_n) that should receive chunks in a round. V^* is initialized by $\{W_n\}$. Afterwards, the selection algorithm, depending on θ , examines three cases using different search algorithms aiming at finding the best algorithm that adds more workers to V^*. After obtaining the three candidate V^* sets, the algorithm chooses the V^* set that produces the minimum makespan. From (4) we compute the makespan as follows
if $\theta = 1$:

$$makespan = \frac{L_{total}}{\sum_{i \in V^*} A_i} \left(\frac{1}{m} \sum_{i \in FS} \frac{S_i}{B_i + S_i} + \frac{B_n}{B_n + S_n} \right) + C \tag{14}$$

where C is a constant

$$C = \sum_{i \in V^*} nLat_i + m.cLat_n \tag{15}$$

if $\theta \neq 1$:

$$makespan = \frac{L_{total}}{\sum_{i \in V^*} A_i} \left(\frac{1-\theta}{1-\theta^m} \sum_{i \in FS} \frac{S_i}{B_i + S_i} + \frac{B_n}{B_n + S_n} \right) + C \tag{16}$$

Now, since

$$lim_{m \to \infty} \frac{1-\theta}{1-\theta^m} = (0 \quad if \quad \theta > 1; \quad (1-\theta) \quad if \quad \theta < 1) \tag{17}$$

and since m (the number of rounds) is usually large (in our experiments, m is in hundreds), we can write:

$$\frac{1-\theta}{1-\theta^m} \approx (0 \quad if \quad \theta > 1; \quad (1-\theta) \quad if \quad \theta < 1) \tag{18}$$

when $\theta > 1$ and by substituting this term into (16) we get

$$makespan = L_{total} \frac{B_n}{(B_n + S_n) \sum_{i \in V^*} \frac{B_i S_i}{B_i + S_i}} \tag{19}$$

when $\theta < 1$ and by substituting the above term into (16) we get

$$makespan = L_{total} \frac{\sum_{i \in V^*} \frac{S_i}{B_i + Si}}{\sum_{i \in V^*} \frac{B_i S_i}{B_i + S_i}} + C \tag{20}$$

Based on the above analysis, we have three selection policies for generating V^*:

- Policy I ($\theta > 1$): this policy aims at reducing the total idle time by progressively increasing the load processed in each round (i.e., $round_{j+1} > round_j$).
- Policy II ($\theta < 1$): this policy aims at maximizing the number of workers that can participate by progressively decreasing the load processed in each round (i.e., $round_{j+1} < round_j$).
- Policy III ($\theta = 1$): this policy keeps the load processed in each round constant (i.e., $round_{j+1} = round_j$)

As shown in Algorithm 1, the three policies are examined in order to choose the one that produces the minimum makespan. In the coming subsections, we will explain the search algorithm adopted by each policy.

4.1 Policy I ($\theta > 1$)

From (19), we can see that under this policy, V^* is the subset that maximizes the sum

$$m_1(V^*) = \sum_{i \in V^*} \frac{B_i S_i}{B_i + S_i} \tag{21}$$

subject to $\theta > 1$ or

$$\sum_{i \in V^*} \frac{S_i}{B_i + S_i} < \frac{B_n}{B_n + S_n} \tag{22}$$

One can observe that this is a Binary Knapsack [6] problem that can be solved using the Brand-and-bound algorithm [6].

4.2 Policy II ($\theta < 1$)

From (20), we can see that under this policy, V^* is the subset that minimizes

$$m_2(V^*) = \frac{\sum_{i \in V^*} \frac{S_i}{B_i + S_i}}{\sum_{i \in V^*} \frac{B_i S_i}{B_i + S_i}} \tag{23}$$

subject to $\theta < 1$ or

$$\sum_{i \in V^*} \frac{S_i}{B_i + S_i} > \frac{B_n}{B_n + S_n} \tag{24}$$

To start with, we should initiate V^* with the first worker, W_0, that minimizes $m_2()$.

Lemma 1. $m_2(V^*)$ is minimum if $V^* = \{W_0\}$ such that $B_0 \geq B_i \forall P_i \in V$.
Proof. Consider an arbitrary subset $X \subseteq V, X = \{P_1, P_2, ...P_r\}$. We have:

$$B_0 > B_i \Rightarrow \sum_{i=1}^{r} \frac{B_0 S_i}{B_i + S_i} > \sum_{i=1}^{r} \frac{B_i S_i}{B_i + S_i} \Rightarrow B_0 \sum_{i=1}^{r} \frac{S_i}{B_i + S_i} > \sum_{i=1}^{r} \frac{B_i S_i}{B_i + S_i}$$

$$\Rightarrow \frac{\frac{S_0}{S_0 + B_0}}{\frac{B_0 S0}{B_0 + S_0}} < \frac{\sum_{i=1}^{r} \frac{S_i}{B_i + S_i}}{\sum_{i=1}^{r} \frac{B_i S_i}{B_i + S_i}} \Rightarrow m_2(V^*) < m_2(X) \quad \forall W_i \in V \tag{25}$$

After adding W_0 to V^*, we should keep conservatively adding more workers until constraint (24) is satisfied. In fact, the next W_k that should be added to V^* is the one that satisfies the following inequality:

$$m_2(V^* \cup \{W_k\}) \leq m_2(V^* \cup \{W_j\}) \forall W_j \in V - V^* \tag{26}$$

The Greedy algorithm described below progressively adds more P_k until V^* satisfies (24), i.e. until ($\theta < 1$). The run time complexity of this search is $O(n)$.

AlgorithmGreedy(V, *thetaCondition*)
Begin
  Search $W_n \in V : B_n/(B_n + S_n)\ B_i/(B_i + S_i) \forall W_i \in V$
  Search $W_0 \in V : B_0 \geq B_i \quad \forall W_i \in V$
  $V^* = \{W_n, W_0\}$; $V = V - V^*$;
  Repeat
    Search worker W_k satisfy $m_2(V^* \cup \{W_k\}) \leq m_2(V^* \cup \{W_j\})\ \forall W_j \in V$
    $V^* = V^* \cup \{W_k\}$; $V = V - \{W_k\}$;
  Until thetaCondition;
  Return (V^*);
End

4.3 Policy III ($\theta = 1$)

Under this policy, we need to find V^* that minimizes the following makespan function

$$m_3(V^*) = \frac{\sum_{i \in V^*} \frac{S_i}{B_i + S_i}}{\sum_{i \in V^*} \frac{B_i S_i}{B_i + S_i}} \tag{27}$$

subject to $\theta = 1$ or

$$\frac{B_n}{(B_n + S_n)\sum_{i \in V^*} \frac{S_i}{B_i + S_i}} = 1 \tag{28}$$

It is noticeable that $m_3()$ is the same as $m_2()$ (Policy II). However, the two objective functions differ with respect to their constraints. Therefore, we can use the same Greedy search algorithm explained earlier with the exception that the termination condition should be $\theta = 1$ (instead of $\theta < 1$).

5 Experimental Results

In order to evaluate our new algorithm, MRRS, we developed a simulator using the SIMGRID toolkit [7], which has been used to evaluate the UMR algorithm. We conducted a number of experiments that aim at i) showing the validity of our approximation assumptions discussed in Section 4, and ii) showing that the MRRS algorithm is superior to its predecessors, namely LP and UMR.

5.1 Validity of Approximation Assumptions

The experiments we conducted show that the absolute deviation between the theoretically computed makespan, as analyzed in Section 4, and the makespan observed through the simulation experiments is negligible. This confirms that the approximation assumptions adopted in our analysis are plausible. The parameters used in our experiments are:

- Number of workers: $N = 50$.
- Total workload: 10^6 flop
- Computation speed: Randomly selected from $[S_{min}, 1.5 \times S_{min}]$, where S_{min} = 50 Mflop/s
- Communication rate (Mbps): Randomly selected from $[0.5 \times N \times S_{min}, 1.5 \times N \times S_{min}]$
- Computation and communication latencies: from 10 to 10^{-2} (s)

Let us denote:

- MK_e is the makespan obtained from the experiments.
- MK_1, MK_2, MK_3 are the makespans computed by formula (6), (8) and (9) respectively.
- $D_i (i = 1, 2, 3)$ is the absolute deviation between the theoretical makespan, MK_i, and the experimental makespan MK_e. Therefore:

$$D_i = 100 * \frac{|MK_i - MK_e|}{MK_e}\% \tag{29}$$

Table 1 summarizes the absolute deviations computed for different latencies. From these results we can make the following remarks:

- The absolute deviation between the theoretical and the experimental makespans ranges from 0.5% to 3.1%, which is negligible.
- We notice that $D_2 < D_1 < D_3$. The justification is that the absolute deviation (D) is propotional to the number of participating workers in a given selection policy. The more workers participate, the larger D becomes. As we recall that D_2 represents the deviation caused by policy II ($\theta > 1$), which is the most conservative policy with respect to the number of workers allowed to participate. D_3 represents the deviation caused by policy III ($\theta < 1$), which is the most relaxed policy with respect to the number of participating workers. D_1 of policy I ($\theta = 1$) falls in the middle with respect to the number of participating workers and according the observed deviation.

Table 1. The Absolute Deviation between the Experimental and Theortical Makespans

nLat, cLat (s)	D1 (%)	D2 (%)	D3 (%)
1	3.15	2.42	3.34
0.1	2.23	1.75	2.27
0.01	1.51	0.92	1.94
0.001	0.82	0.51	1.25

5.2 Comparison with Previous Algorithms

We compare MRRS with the most powerful scheduling algorithm, namely UMR [4] and LP [3]. The performance of these algorithms have been compared with respect to three metrics:

- The normalized makespan, that is normalized to the run time achieved by the best algorithm in a given experiment;
- The rank which ranges from 0 (best) to 2 (worst);
- The degradation from the best, which measures the relative difference, as a percentage, between the makespan achieved by a given algorithm and the makespan achieved by the best one.

These metrics are commonly used in the literature for comparing scheduling algorithms [4,3]. The results summarized in Table 2 suggests that MRRS can outperform its competitors in most of the cases. MRRS's rank dropped from the best to the second in 12% of the cases with 5.4% performance degradation in comparision with the UMR.

Figure 2 shows that UMR has chances of outperforming MRRS only if the number of workers is small ($n \leq 20$). In this case, the worker selection module of MRRS does not have enough workers, which denies the MRRS from using one of the worker selection policies, namely Policy II. LP has almost no chance to win. This is due to the fact that LP does not have any effective strategy of reducing the idle time of workers at the end of each round.

Table 2. Performance comparisons among MRRS, UMR and LP Algorithms

Algorithm	Normalized Makespan	Rank	Degradation from the best
MRRS	1	0.12	0.65
UMR	1.21	0.88	21.4
LP	1.59	2	59.8

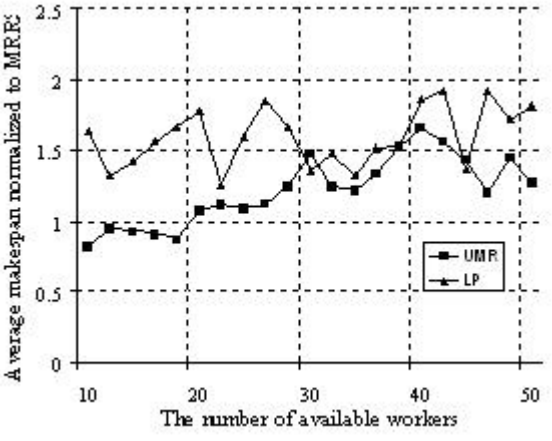


Fig. 2. The relation between n and the makespan

6 Conclusion

The ultimate goal of any scheduling algoritm is to minimize the makespan. UMR and LP are among these algorithms that have been designed to schedule divisible loads in heterogeneous environments where workers have different CPU speeds connected to links with different bandwidths. However, these algorithms do not take into account a number of chief parameters such as bandwidths and the inevitable latencies of communication and computation. Furthermore, present algorithms are not equipped with a resource selection mechanism as they assume that all available workers will participate in processing the workload. In this work, we presented the MRRS algorithm that divides the workload into chunks in light of more realistic parameters mentioned earlier. We explained how the MRRS has a worker selection with an asymptotic optimality. To the best of our knowledge, MRRS is the first scheduling algorithm that addresses the worker selection problem. The simulation experiments show that MRRS is superior to its predecessors especially when it is put into operation in a colossal computing platform such as the Grid, which agglomerates an abundant pool of heterogeneous workers.

Acknowledgements

Our research is supported by the "Fostering Talent in Emergent Research Fields" program, sponsored by the Ministry of Education, Culture, Sports, Science and

Technology, Japan. This work has been also funded by research grant #02-06-9-11/06 from the Scientific Research Council of the UAE University, UAE.

References

1. Bharadwaj, V., Ghose, D., Mani, V., Robertazzi, T.G.: Scheduling Divisible Loads in Parallel and Distributed Systems. IEEE Computer Society Press, Los Alamitos (1996)
2. Foster, I., Kesselman, C.: Grid2: Blueprint for a New Computing Infrastructure, 2nd edn. Morgan Kaufmann Publisher, San Francisco (2003)
3. Beaumont, O., Casanova, H., Legrand, A., Robert, Y., Yang, Y.: Scheduling Divisible Loads on Star and Tree Networks: Results and Open Problems. IEEE Transactions on Parallel and Distributed Systems 16(3), 207–218 (2005)
4. Yang, Y., Raart, K.V., Casanova, H.: Multiround Algorithms for Scheduling Divisible Loads. IEEE Transaction on Parallel and Distributed Systems 16(11), 1092–1104 (2005)
5. Bertsekas, D.P.: Constrained Optimization and Lagrange Multiplier Methods. Athena Scientific, Belmont (1996)
6. Martello, S., Toth, P.: Knapsack problems: algorithms and computer implementations. Wiley, Chichester (1990)
7. Casanova, H.: Simgrid: a Toolkit for the Simulation of Application Scheduling. In: Proc. of the IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2001), Australia, pp. 430–437 (2001)

DHT-Based Self-adapting Replication Protocol for Achieving High Data Availability*

Predrag Knežević[1], Andreas Wombacher[2], and Thomas Risse[1]

[1] Fraunhofer IPSI
Dolivostrasse 15, 64293 Darmstadt
Germany
{knezevic,risse}@ipsi.fhg.de
[2] University of Twente
Department of Computer Science
Enschede, The Netherlands
a.wombacher@cs.utwente.nl

Abstract. An essential issue in peer-to-peer data management is to keep data highly available all the time. The paper presents a replication protocol that adjusts autonomously the number of replicas to deliver a configured data availability guarantee. The protocol is based on a Distributed Hash Table (DHT), measurement of peer online probability in the system, and adjustment of the number of replicas accordingly. The evaluation shows that we are able to maintain the requested data availability achieving near to optimal storage costs, independent of the number of replicas used during the initialization of the system.

Keywords: decentralized data management, distributed hash tables, self-adaptation, high data availability.

1 Introduction

Distributed Hash Tables provide an efficient way for storing and finding data within a peer-to-peer network, but they do not offer any guarantees about data availability. When a peer goes offline, the locally stored data become inaccessible as well. The work presented in the paper extends existing Distributed Hash Tables (DHTs) by delivering a configurable data availability. The protocol is fully decentralized and has the following features: (1) measuring current average peer online probability in the community, (2) computing the number of replicas that every stored object needs to have in order to achieve the requested average data availability locally, and (3) adjusting the number of needed replicas for locally stored items so that the requested data availability is achieved and kept.

The research presented in this paper is motivated by the BRICKS[1] project, which aims to design, develop and maintain a user and service-oriented space of digital libraries on the top of a decentralized/peer-to-peer architecture. During the run-time, the

* This work is partly funded by the European Commission under BRICKS (IST 507457).

[1] BRICKS - Building Resources for Integrated Cultural Knowledge Services, http://www.brickscommunity.org

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 201–210, 2009.

system requires to have a variety of system metadata like service descriptions, administrative information about collections, or ontologies globally available [1].

The paper is organized in the following way. The next Section gives details about the problem. Section 3 describes the applied replication technique. The approach is evaluated by using custom-made simulator and various results are presented in Section 4. Related work to the idea presented in the paper is given in Section 5. Finally, Section 6 gives conclusions and some ideas for the future work.

2 Problem Statement

As suggested in [2], a common DHT API should contain at least the following methods: (1) **route(Key, Message)** - routes deterministically forward a message according to the given key to the final destination; (2) **store(Key, Value)** - stores a value under the given key in DHT; and (3) **lookup(Key)** - returns the value associated with the key.

Every peer is responsible for a portion of the key space, so whenever a peer issues a **store** or **lookup** request, it will end up on the peer responsible for that key. Although implementation specific, a peer is responsible for a part of the keyspace nearest to its ID. The distance function can be defined in many ways, but usually it is simple arithmetic distance. When the system topology is changed, i.e. a peer goes offline, some other online peers will be now responsible for the keyspace that has belonged to the offline peer. The part of the keyspace belonging to the currently offline peer will be split and taken over by peers that are currently nearest to keys in the part of the keyspace. Also, peers joining the system will take responsibility for a part of the keyspace that has been under control of other peers until that moment.

As it is already mentioned, our decentralized storage uses a DHT layer for managing XML data. Unfortunately, all DHT implementations do not guarantee data availability, i.e. when a peer is offline, then locally stored data are offline too. Therefore, enabling high data availability can be achieved by wrapping a DHT and adding the implementation of a replication protocol on top of it, and at the same time being transparent to the users, i.e. providing the same API to upper layers as a DHT does.

3 Approach

Every value and its replicas are associated with a key used for **store** and **lookup** operation. The first replica key is generated using a random number generator. All other replica keys are correlated with the first one, i.e. they are derived from it by using the following rule:

$$replicaKey(ron) = \begin{cases} c & : \quad ron = 1 \\ hash(replicaKey(1) + ron) & : \quad ron \geq 2 \end{cases} \tag{1}$$

where ron is a replica ordinary number, c is a random byte array, $hash$ is a hash function with a low collision probability. $replicaKey$ and i are observed as byte arrays, and $+$ is an array concatenation function.

The above key generation schema ensures that all replicas are stored under different keys in the DHT. Further, two consecutive replica keys are usually very distant in the

key space (a consequence of using a hash function) meaning that replicas should be stored on different peers in a fairly large network. Based on the key generation schema in Formula 1, deriving the key of the ron^{th} replica requires access to the first replica key and the replica ordinary number (ron). Therefore, this information is wrapped together with the stored value in an instance of `Entry` class.

Since the wrapper around DHT implements the common DHT API, **store** and **lookup** operations must be re-implemented.

store(Key, Value) When a value is created, it is wrapped in a number of instances of `Entry` class and appropriate keys are generated. The constructed objects are stored using the basic DHT store operation. If the objects already exist, their old value is overwritten.

lookup(Key) When a peer wants to get a value, it is sufficient to return any available replica. Therefore, the keys for the assumed number of replicas are generated and the basic DHT lookup operation is applied until the corresponding `Entry` object is retrieved.

When a peer **rejoins** the community, it does not change its ID, and corollary the peer will be now responsible for a part of the keyspace that intersects with the previously managed keyspace. Therefore, the peer keeps previously stored data, but no explicit data synchronization with other peers is required. Replicas, whose keys are not anymore in the part of the keyspace managed by the rejoined peer, can be removed and may be sent to peers that should manage them.

3.1 Number of Replicas

In order to add data availability feature to the existing DHT, every stored value must be replicated R number of times. The number of replicas R is independent of objects, i.e. the same data availability is requested for all objects. Joining the community for the first time, a peer can assume an initial value for R, or can obtain it from its neighbors.

To meet the requested data availability in a DHT, the number of replicas of stored data at each peer has to be adjusted. Therefore, every peer measures once per online session, i.e. a peer being online, the current average peer online probability, and knowing the requested data availability, calculates the new value for the number of replicas R. By knowing the previous value R', a peer removes removes the replicas with ordinary number ron greater than R from its local storage replicas, if it turns out that less replicas are needed than before ($R < R'$). In case a higher number of replicas are needed ($R > R'$), a peer creates new replicas of the data in its local storage under the keys $replicaKey(j), j = R' + 1, ..., R$.

Be aware that replicating all R replicas (in case of $R > R'$) instead of replicas , $R' + 1, \ldots, R$ as originally proposed in [3] results in wrong online probability measures as discussed in the next section.

Having the peer online probability p and knowing the requested data availability a, a peer can compute the needed number of replicas. Let us denote with Y a random variable that represents the number of replicas being online, and $P(Y \geq y)$ is the probability that at least y replicas are online. Under an assumption that peers are independent and behave similarly, the probability follows Binomial distribution. The probability a that a value is available in a DHT with peer online probability p is equal to the probability that at least one replica is online ($P(Y \geq 1)$):

$$a = P(Y \geq 1) = 1 - (1 - p)^R \tag{2}$$

Therefore, the number of needed replicas R is

$$R = \left\lceil \frac{\log(1 - a)}{\log(1 - p)} \right\rceil \tag{3}$$

3.2 Measuring Peer Availability

Measuring the average peer online probability is based on probing peers and determining the peer online probability as a ratio between the number of positive and total probes. However, the measuring cannot be done directly (i.e. pinging peers) because we do not know anything about the peer community, i.e. peer IDs, and/or their IP addresses. Our wrapper uses the underlying DHT via the common DHT API, i.e. its routing operation, for probing the availability of replicas. According to the key generation properties (Formula 1) stated before, we assume that every replica of a particular object is stored on a different peer within a fairly large network. Based on this assumption, the peer online probability equals the replica online probability. Be aware that this condition is only valid if each ron^{th} replica is stored in a single online or offline peer. Having several ron^{th} replicas distributed over several online or offline peers results in peer online probability measurement errors, because the availability of a replica does no longer equal the availability of a single, but of several peers. Having only one replica under a key is supported by the implemented replication, and therefore the measurement of peer online probability can be realized as follows. Every peer has some replicas in its local store and based on that and Formula 1, all keys of all other replicas corresponding to the same objects can be generated, and their availability can be checked.

Of course, we are not in a position to check all replicas stored in the system. In order to optimize the number of probes, we consider the confidence interval theory [4] to find out what is the minimal number of replicas that has to be checked, so the computed average replica online probability is accurate with some degree of confidence.

Unfortunately, the direct usage of the confidence interval theory is not so promising for practical deployment. Namely, by requesting a small absolute error (e.g. $\delta = 0.03$), and high probability of having the measurement in the given interval (e.g. $C = 95\%$), a peer should make at least 1068 random replica availability probes, generating high communication costs. Even if the network is not so large (e.g. 400 peers), the number of probes n drops only to 290.

Therefore, we need to modify slightly our probing strategy to achieve the requested precision. After probing some replicas using their keys, and calculating the peer availability, the peer uses the same replica keys to ask other peers about their calculated value. Finally, the peer computes the average peer online availability as average of all received values. This is done by routing an instance of $PeerProbabilityRequest$ message using the **route** DHT method with all replica keys from the probe set. The contacted peers route their answers ($PeerProbabilityResponse$) back. The number of generated probe messages is now $2n$, but we are able to check up to n^2 replicas. Measuring can be done with low communication costs; already with $n = 33$, and 66

messages we are able to check up to 1089 replicas, and achieve good precision in a network of any size.

3.3 Costs

In general, the total costs consists of two major parts: communication and storage costs. The presented approach is self-adaptable; it tries to reach and keep the requested data availability with minimum costs at any point in time. If the initial number of replicas is not sufficient to ensure the requested data availability, new replicas will be created, i.e. storage costs will be increased. On the other hand, if there are more replicas than needed, peers will remove some of them, reducing storage costs.

Let us denote with $S(t)$ **average storage costs** per peer in a system with M objects and N peers that are online with probability p. The minimum storage costs per peer S_{min} that guarantee the requested data availability a is:

$$S_{min} = \frac{M}{N} R \tag{4}$$

where R is computed like in Formula 3.

Every **store** operation generates R $StoreRequest$ and R $StoreResponse$ messages. A **lookup** operation does not have fixed costs, it stops sending $LookupRequest$ messages when a replica is found. Sometime, already the 1^{st} replica is available, and only two messages ($LookupRequest$ and $LookupResponse$) are generated. In an extreme case, $2R$ messages must be generated in order to figure out if the requested object is available or not. It can be shown that on average a **lookup** needs to generate R messages, before finding a replica of the requested object.

The proposed approach introduces additional **communication costs** generated by measuring peer online probability p and adjusting the number of replicas. Every measurement generates n $LookupRequest$, n $LookupResponse$, n $PeerAvailabilityRequest$, and n $PeerAvailabilityResponse$ messages, where n is the number of replicas to probe. As stated in the Section 3.2, n depends on the absolute error δ we want to allow, and the probability that the measured value is within the given interval $(p - \delta, p + \delta)$. If the computed number of replicas R is greater from the previous one R', the peer will create additionally $(R - R')S(t)$ $StoreRequest$ messages in average.

4 Evaluation

The evaluation has been done using a custom-made simulator. The protocol implementation is based on FreePastry [5], an open-source DHT Pastry [6] implementation. The obtained results match to the developed model, i.e. by using the computed number of replicas and the defined operations. The DHT achieves the requested data availability.

4.1 Settings

Every simulation starts with the creation of a DHT community, populating it with a number of entries, which are initially replicated a predefined number of times. During

this phase, all peers are online, to ensure that the load of the local peer storages is a balanced.

After initializing the DHT, the simulator executes a number of iterations. In every iteration, peers, whose session time is over go offline, and some other peers with a probability p come online and stay online for some time. The session length is generated by using a Poisson distribution, with the given average session length λ.

Every time when a peer comes online, it measures the average peer online probability, determines the number of required replicas, and tries once to adjust the number of replicas for locally stored data according to the operations defined in Section 3.2.

Finally, at the end of every iteration, the simulator measures the actual data availability in the system by trying to access all stored data. It iterates over the object keys, picks up an online peer randomly, which then issues a **lookup** request.

The configuration parameters of the simulator consists of the community size N, the requested data availability a, the average peer online probability p, the average session time λ, the initial number of replicas R, and the number of replicas to probe n. These parameters offer a huge number of possible scenarios to simulate, and since the simulations are time-consuming, we have fixed the community size N to 400 peers, the average session time λ to three iterations, and the number of replicas to probe n to 30. As stated in Section 3.2, for the defined network, $n = 17$ would be enough, but we wanted to be on the safe side at the first glance.

The system is initialized with less replicas than needed for reaching the requested data availability. Therefore, peers must cope with that, and create more replicas until the data availability is achieved. The protocol is evaluated both on low and high peer availability network 10 times, and the obtained values are averaged.

4.2 Low Available DHTs

We have set up a DHT with the average peer online probability of 20%, and tried to **reach the data availability** of 99% initializing peers with the number of replicas $R = 5$. After 10 runs, the obtained values are averaged, and the results are shown on Figure 1.

The requested data availability has been reached after the major number of peers has been able to measure the actual peer availability and to create a correct number of replicas (Figure 1a). Figure 1b shows the development of the average error (difference between the requested and obtained data availability) during the simulations. As it can be seen, the DHT maintains the data availability that is a bit under the requested one.

Figure 1c shows the average storage costs per peer during the simulations. After reaching the requested data availability, the storage load stabilizes as well, and matches quite well with the curve "theoretical minimum" (Formula 4 and 3).

4.3 High Available DHTs

The evaluation has been applied also to a DHT with the average peer availability of 50%. The peer has been initialized again with the lower number of replicas $R = 3$, which is insufficient to guarantee the requested data availability of 99%. The obtained results are presented on Figure 2.

The data availability has been reached even faster (Figure 2a) than in the previous case. After measuring the actual peer availability and computing the number of replicas,

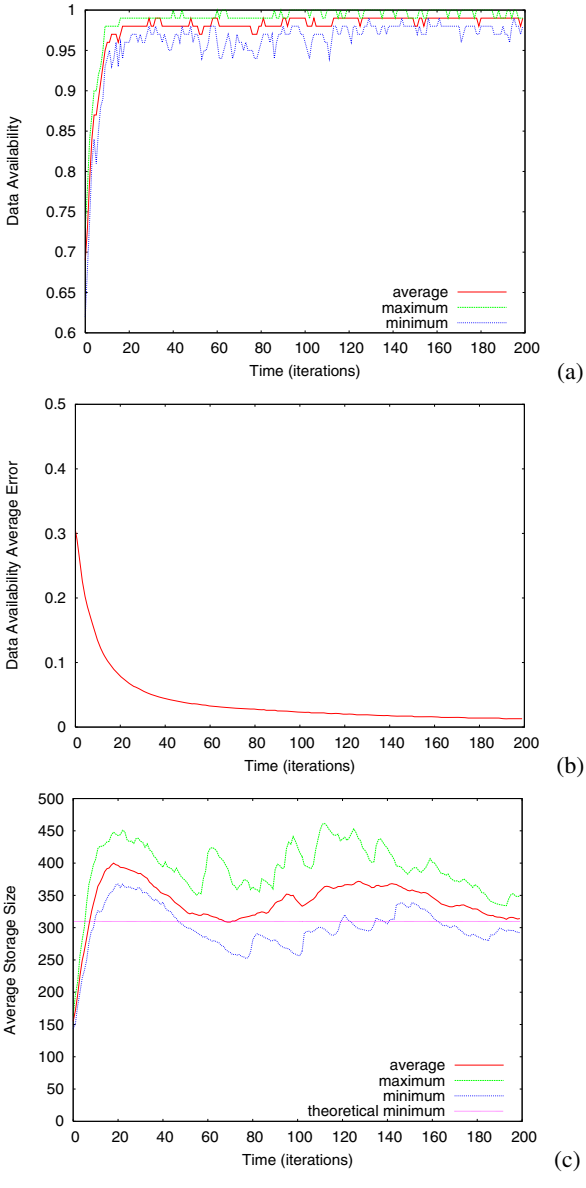


Fig. 1. Reaching the requested availability in networks with 20% peer availability

peers can adjust the number of replicas for all locally stored data. Since more peers are always online, replica adjustment happens to more data in average and the requested data availability is reached faster. The achieved average error (Figure 2b) is as low as in the previous case. Also, the average storage costs (Figure 2c) fit again very well to the developed model.

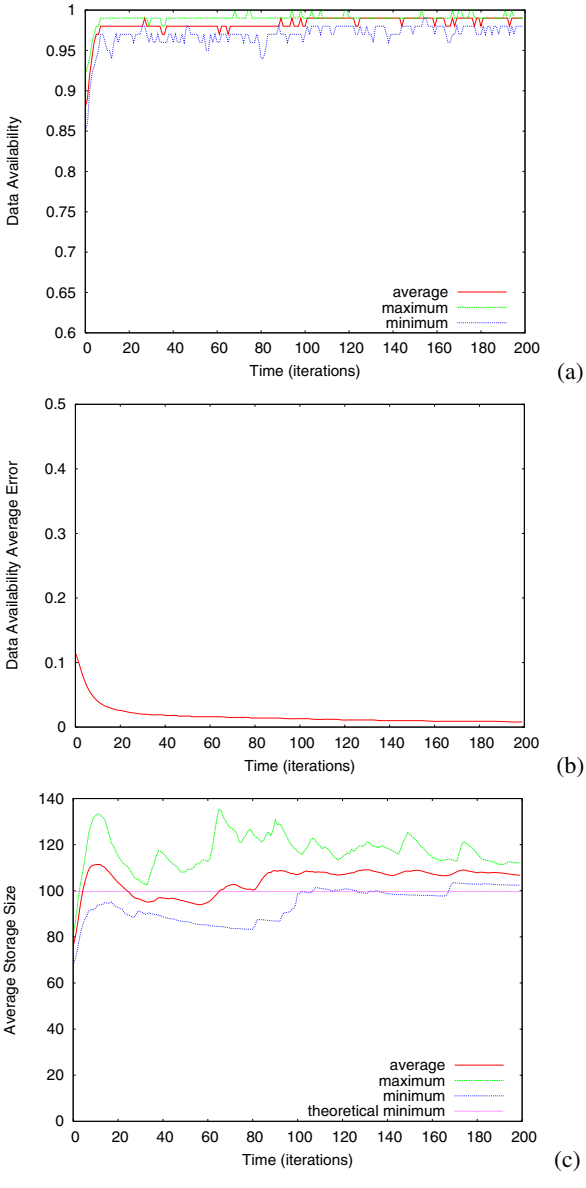


Fig. 2. Reaching the requested availability in networks with 50% peer availability

5 Related Work

P2P file-storing system like CFS [7] and PAST [8] have recognized the need for replication. CFS splits every file in a number of blocks that are then replicated a fixed number of times. PAST applies the same protocol, without chunking files into blocks. The number of replicas must be carefully chosen for the given community, because systems are

not able to correct it, if the requested availability cannot be obtained. Also, if the system behaves much better, local peer storages will contain unnecessary replicas.

Oceanstore [9] is a P2P file-storing system trying to provide a wide-area storage system. Oceanstore is not fully decentralized, it makes a distinction between clients and powerful, highly available servers (i.e. super-peer network). Within the storage, the data are replicated by using erasure coding (Reed-Solomon codes). Erasure coding is also used in TotalRecall [10]. Erasure coding offers lower storage costs compared to replication, if managed data are large in size. However, our DHT layer handles smaller data items. Additionally, if locally stored data are encoded, then it is not possible to process them without retrieving all needed pieces from the network, which would decrease the system performances.

The protocol presented in [11] exploits as well the erasure coding for achieving data availability. Periodically, peers detect current file availability and if below the requested one, replicate randomly files attempting to increase their availability. The idea behind the protocol is similar to our approach, but requires existence of a global index and global directory, where all information about files (locations and availability) are managed. The protocol presented in our paper is fully decentralized, and does not need such data structure for proper functioning.

6 Conclusion and Future Work

The paper has presented a fully decentralized replication protocol suited for DHT networks. It enables reaching and keeping the requested data availability, and at the same time, the protocol keeps storage costs very close to the minimum. The clear advantage of a DHT network that supports the presented protocol is deployment in an environment, whose parameters like peer availability, are not well-known or cannot be good predicted. The system itself will find the optimal number of replicas, and maintain the requested data availability. If the system properties change afterwards, the system will adapt automatically, i.e. no system restart and/or reconfiguration is needed.

Future work will evaluate the robustness of the approach and update issues. In particular, it has to be investigated how the protocol behaves in networks with high churn rate, i.e. peer online/offline rates changes significantly in some periods of system runtime. Since not all peers are always online, an update might not be able to change all replicas, leaving some of them unmodified. Also, uncoordinated concurrent updates of an object result in unpredictable values of object replicas. The protocol presented in the paper has been already extended in order to support solving the above issues [12], and the first analysis shows that we are able to provide data consistency with arbitrary high probabilistic guarantees. The protocol will be evaluated under various system settings and similar communities like in this paper.

References

1. Risse, T., Knežević, P.: A self-organizing data store for large scale distributed infrastructures. In: International Workshop on Self-Managing Database Systems(SMDB), April 8-9 (2005)
2. Dabek, F., Zhao, B., Druschel, P., Stoica, I.: Towards a common api for structured peer-to-peer overlays. In: 2nd International Workshop on Peer-to-Peer Systems (February 2003)

3. Knežević, P., Wombacher, A., Risse, T., Fankhauser, P.: Enabling high data availability in a DHT. In: Grid and Peer-to-Peer Computing Impacts on Large Scale Heterogeneous Distributed Database Systems (GLOBE 2005) (2005)
4. Berry, D.A., Lindgren, B.W.: Statistics: Theory and Methods. Duxbury Press (1995)
5. University, R.: FreePastry - Open-source Pastry Implementation (2006), http://freepastry.org/FreePastry/
6. Rowstron, A., Druschel, P.: Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In: Guerraoui, R. (ed.) Middleware 2001. LNCS, vol. 2218, p. 329. Springer, Heidelberg (2001)
7. Dabek, F., Kaashoek, M.F., Karger, D., Morris, R., Stoica, I.: Wide-area cooperative storage with CFS. In: Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles, pp. 202–215. ACM Press, New York (2001)
8. Rowstron, A., Druschel, P.: Storage management and caching in past, a large-scale, persistent peer-to-peer storage utility. In: Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles, pp. 188–201. ACM Press, New York (2001)
9. Kubiatowicz, J., Bindel, D., Chen, Y., Eaton, P., Geels, D., Gummadi, R., Rhea, S., Weatherspoon, H., Weimer, W., Wells, C., Zhao, B.: Oceanstore: An architecture for global-scale persistent storage. In: Proceedings of ACM ASPLOS. ACM, New York (2000)
10. Bhagwan, R., Tati, K., Cheng, Y.-C., Savage, S., Voelker, G.M.: Total recall: System support for automated availability management. In: First ACM/Usenix Symposium on Networked Systems Design and Implementation, March 29-31, 2004, pp. 337–350 (2004)
11. Cuenca-Acuna, F.M., Martin, R.P., Nguyen, T.D.: Autonomous replication for high availability in unstructured p2p systems. srds 00, 99 (2003)
12. Knežević, P., Wombacher, A., Risse, T.: Highly available DHTs: Keeping data consistency after updates. In: Proceedings of International Workshop on Agents and Peer-to-Peer Computing (AP2PC) (2005)

A Hierarchical n-Grams Extraction Approach for Classification Problem

Faouzi Mhamdi[1], Ricco Rakotomalala[2], and Mourad Elloumi[1]

[1] UTIC, Unité de recherche en Technologies de l'Information et de la Communication, École Supérieure des Sciences et Techniques de Tunis, Tunisie
faouzi.mhamdi@ensi.rnu.tn, mourad.elloumi@fsegt.rnu.tn
[2] Laboratoire ERIC, Université Lyon 2, France
ricco.rakotomalala@univ-lyon2.fr

Abstract. We are interested in protein classification based on their primary structures. The goal is to automatically classify proteins sequences according to their families. This task goes through the extraction of a set of descriptors that we present to the supervised learning algorithms. There are many types of descriptors used in the literature. The most popular one is the n-gram. It corresponds to a series of characters of n-length. The standard approach of the n-grams consists in setting first the parameter n, extracting the corresponding n-grams descriptors, and in working with this value during the whole data mining process. In this paper, we propose an hierarchical approach to the n-grams construction. The goal is to obtain descriptors of varying length for a better characterization of the protein families. This approach tries to answer to the domain knowledge of the biologists. The patterns, which characterize the proteins' family, have most of the time a various length. Our idea is to transpose the frequent itemsets extraction principle, mainly used for the association rule mining, in the n-grams extraction for protein classification context. The experimentation shows that the new approach is consistent with the biological reality and has the same accuracy of the standard approach.

Keywords: Data mining, Protein Classification, SVM, Association Rules, Frequent itemsets, n-grams.

1 Introduction

Biologists play a central role in the classification of individuals such as animals, plants, genes, proteins, etc. However, the great amount of biological data such as proteins, DNA, RNAm etc. involves a strong need for the intervention of other research tools and techniques in order to help these biologists, mainly because the manual classification has become almost impossible.

Nowadays, Computer Science is the most requested tool. From the cooperation between computer scientists and biologists resulted a great number of disciplines specialised for the manipulation and analysis of biological data such as Bioinformatics and Biomining. The majority of these disciplines are inserted into the process of the knowledge discovery from databases [1]. There are many techniques and methods for

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 211–222, 2009.

the comparison and the classification of biological sequences such as BLAST, FASTA. They are based on the computation of the similarities between the sequences. For instance, Smith and Waterman developed a technique based on the dynamic programming. These approaches reach to matrixes of scores such as PAM and BLOSUM. Some techniques are based on alignments between the sequences [2], whereas others are based on the hidden Markov Model: HMM [3], SAM[14] or HMMER[15]. Recently, several data mining techniques are used for the supervised classification. They are applied in the biological sequence analysis and especially in the protein classification. Various supervised learning algorithms are available (e.g. neural networks, nearest neighbours, decisions trees, etc.), the most popular one in the protein classification domain is the support vector machine (SVM) [16].

In this paper, we outline a new approach which is better consistent with the biological domain knowledge. The biological reality says that the protein families are characterized with patterns of different length. Biologists identify each protein family with a specific field. A field is a set of motifs that are dispersed throughout sequences of a given family.

Until now, we set the length of the motifs before the whole classification process. It is the principle of the n-grams descriptor extraction. In this case, the length n is necessarily a compromise between several constraints such as computational capability, the kind of relevant information captured, the number of obtained descriptors, etc. This constraint does not correspond to the biological domain knowledge. The idea is thus to go past this constraint by the construction of descriptors with varying length without significantly increasing the computational complexity. In this perspective, we develop a hierarchical approach where the length of the achieved descriptors is not constrained. In order to evaluate our method, on the one hand, we compare our results to a standard n-gram descriptor extraction where we set first the value of n to 3 (3-grams); and on the other hand, we compared our results to a trivial approach where we extract and set together all the n-grams with various values of n (n = 1, 2, etc.).

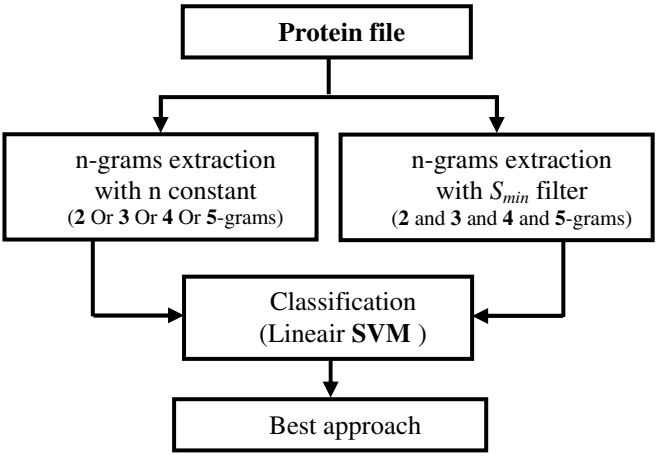


Fig. 1. Evaluation process of the tow n-grams construction approaches

This paper is organised as follows: In the second section, we present the protein classification problem. We outline the n-grams approach, especially the descriptor extraction and the construction of the learning set. In the third section, we explain the hierarchical approach. Experiments and results are reported in the fourth section. We discuss these results. We conclude in the fifth and last section of this paper

2 The Protein Classification Problem

The protein classification is one of the most important tasks of biologists. We want to propose a framework where the classification process relies mainly on the primary description of the proteins. A protein consists of amino acids. There are 20 kinds of amino acids. A protein sequence is thus a set of amino acids that have a given length.

Our process refers to the knowledge discovery process. The steps are well identified by Fayyad and al. [1]. We transpose the knowledge discovery in databases process into a knowledge discovery from biological data. This process involves three steps: the data preparation, including descriptor extraction and data cleaning; the data mining phase where we use the various learning algorithms, a supervised learning algorithm in our context; the validation and the deployment of the classifier e.g. classifying a protein into their family.

2.1 Data Preparation

This step consists in selecting protein families and extracting the sequences from a data bank (SCOP). This step requires the intervention of a biologist, which is the domain expert. Then, we gather these sequences in files by grouping them according to their family membership, see figure 2.

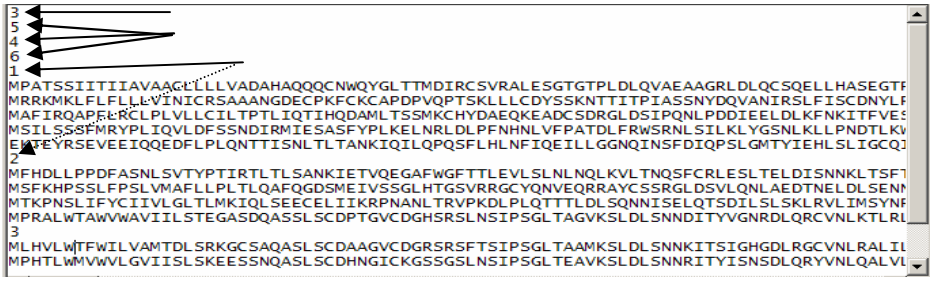


Fig. 2. Protein dataset

2.2 N-Grams Extraction

A protein sequence consists of a set of ordered amino acids. From a certain point of view, an amino acid can be considered as a character and a protein as a text. We then use text mining approach (text categorization approach) to extract descriptors [5]. These descriptors help us build a data table "proteins × descriptors". The supervised learning algorithms are executed on this dataset.

Among the descriptor extraction techniques, we are especially interested in the n-grams extraction. A n-gram is a sub-sequence of *n* characters from a given sequence

of characters. For any sequence, the n-grams set is obtained by moving a window of n-characters on the entire sequence . This moving is performed character by character.

In each moving, the sub-sequence of n amino-acids is extracted. The set of these sub-sequences build the n-grams that can be deduced from a sole sequence. This process will be repeated for all sequences. The algorithmic complexity of these n-grams extraction process is in $O(m \times n \times p)$ with n the size of n-gram, m the size of a sequence and p the number of sequences.

2.3 Data Table Construction

The obtained dataset T can have different forms, especially about the weighting values. The weighting consists in defining the way of filling out the table, which means the affected value $T(i,j)$ where i represents the i^{th} sequence and j represents the j^{th} descriptor or n-gram. There are many weighting types in literature. We can for instance cite:

- Boolean weighting: indicates whether a n-gram is present within a sequence or no.
- Occurrence weighting : indicates the number of occurrences of a n-gram within a sequence.
- Frequency weighting : indicates the relative frequency of a n-gram according to the number of 3-grams composing a sequence.
- TF*IDF weighting: corrects the 3 -grams frequency according to its frequency in a file.

In our context, we adopt a boolean weighting [7]. Our dataset will thus be a boolean table where 1 means the n-gram presence within a sequence, 0 its absence(Figure 4).

2.4 The Drawbacks of This Approach

In the first approach, the standard approach, we define at first the length “n” of the n-gram descriptors used in the subsequent learning algorithm. There are some drawbacks here:

- The best value “n” is not obvious. The chosen value is in reality a compromise, if "n" is too small, the retrieved information for the classification is often too poor; if "n" is too large, we obtain very specific descriptors, which are not useful for the majority of families.

- The number of potential descriptors grows exponentially with the length "n" of the n-grams. If we have 20 amino acids, the possible 2-grams number are $20^2 = 400$ and the possible 3-grams number are $20^3=8000$. Usually, the n-grams number is equal to 20^n . Two problems arise from this potential large number of descriptors: the computational time and the memory usage become a constraint; the resulting classifiers are affected by the "curse of dimensionality", they overfit the learning set when the ratio between the number of descriptors and the number of examples is too large, they generalize poorly when we want to classify a new protein.

- The a priori value "n" may fit to discriminate some families but totally inadequate for other families. In this point of view, setting n as a parameter is not compatible with the biological reality.

As a trivial solution and to overcome these drawbacks, we can extract all n-grams with (n = 2, 3, 4, 5) lengths and use them to classify the proteins. We restrict "n" to 5

because, beyond this value, the computation is not possible on our Personal Computer. Indeed, if we count the descriptors extracted according to this trivial approach, we obtain in average about 72000 descriptors. Even if the computation is possible, learning a classifier in this very large representation space on a hundred or so examples is not realistic.

Between these two solutions, we propose an heuristic approach where the length of the descriptors is an outcome of the features' extraction algorithm. Both the computational time and the number of extracted features must be reasonable. In the following section, we present the hierarchical approach and we compare it to the others.

3 The Hierarchical Approach of Descriptors Extraction

In this section, we describes the hierarchical approach of the descriptors extraction, we compare it with the trivial approach.

3.1 Hierarchical Principle

The hierarchical approach consists in building n-grams with different lengths. This construction is carried out in a hierarchical way, which means we extract (n+1)-grams descriptors from n-grams. It is ascending because the initial step is the extraction of the “valid” 2-grams, then we detect the “valid” 3-grams from these 2-grams, etc.

The key point of the process is the definition of the "valid" term. If all of the extracted 2-grams are valid without restriction, and all deducted 3-grams are valid, and so on, this corresponds to the trivial method where we get together all the n-grams with different values of "n". In our context, we use the frequent itemsets principle suggested by the association rule extraction algorithm. We define a minimum support S_{min} to filter the n-grams at each step. A n-gram is “valid” if the relative number of its apparition (i.e. the ratio between the number of proteins where it appears and the total number of proteins in the dataset) is larger than the minimum support. This restriction enables to master the amount of computation, we are confident with this assertion.

But we hope also, this is less certain, only the experiments allow checking this one, which enables to remove the irrelevant (because they are too infrequent) descriptors. Figure 3 describes the n-grams hierarchical construction principle.

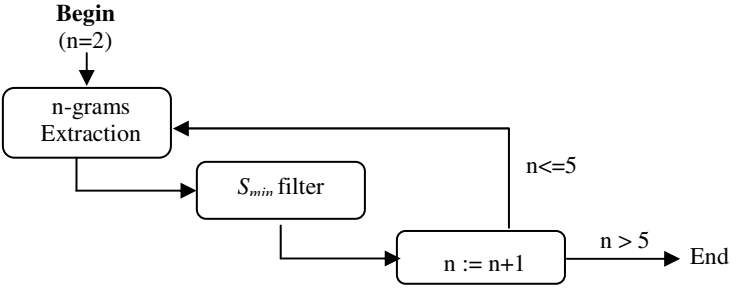


Fig. 3. n-grams hierarchical construction process

3.2 Frequent Itemsets

Mining association rules is a privileged topic of the knowledge extraction from the data. The A-PRIORI algorithm based on the itemset support and the rules confidence is an efficient solution for the rule extraction problems [9]. The approach support/confidence consists in searching for association rules whose support and confidence go beyond fixed threshold by the user in preconditions, namely S_{min} and $Conf_{min}$. Here is an example taken from the marketing domain. Given R an association rule, $\boldsymbol{R : X \rightarrow Y}$ *(A%, B%)* where *A%* points out that X and Y are present together in A% of the transactions (the support for the rule); and B% the customers who bought X have also bought Y (the confidence for the rule).

The rule extraction algorithm, which is based on the support-confidence checking, go through the trellis of itemsets to look for the frequent itemsets, of which support goes beyond S_{min} [8]. *A-PRIORI* is the most popular algorithm [9], it mainly consists of two main steps: We search the frequent itemsets i.e. the itemsets (a set of items, a set of product in the marketing domain, a n-grams in our protein classification context) of which support goes beyond S_{min} by sweeping up the trellis of itemsets within its width and by computing the frequencies with a counter within a base. This method requires a scan of the whole database for each trellis level.

For each frequent itemset X, we keep only the rules that have $X / Y \rightarrow Y$ as a type, with $Y \subset X$ of which confidence goes beyond the threshold $Conf_{min}$.

3.3 Frequent n-Grams

In our context, we are interested in the first phase of the previous algorithm, that is, look for the frequent itemsets. We are going to adapt this approach in order to extract n-grams of varying length and which are frequent. The only parameter of the algorithm is the minimum support S_{min}: in the n-grams extraction phase, we remove the n-grams that have a frequency lower than S_{min}. The number of the descriptors and the length of each descriptor (n-gram) are outcomes of the algorithms.

We can thus present our problem in the following way. To discriminate two families F1 and F2, a relevant descriptor must be frequent in the first family and infrequent in the second one. When the descriptor is frequent for both families, it is not relevant for the classification task. For this reason, the minimum support S_{min} is defined on the families and not on the whole dataset. The question is: which is the family of reference used when we define the S_{min} parameter? A simple example enables to understand the alternative situations.

Take the example of the figure 3, let's take F1 and F2 two protein families, where N1 is the number of F1 sequences and N2 is the number of F2 sequences. We have N1=9, N2=4 and N1+N2=13. We set S_{min} =50%. This value can be used in three different ways:

- 50% according to the file. In this case, we just save the n-grams that exist at least in 50% of the sequences, which means those that have a frequency >= 4.5. As a result, the set of saved n-grams is {ng1, ng2, ng4}.
- 50% according to the most frequent family. In this case, we just save the n-grams that exist at least in 50% of F1. This refer to those that have a frequency >= 6.5. As a result, the set of saved n-grams is {ng1, ng2, ng4, ng5}.

	ng1	ng2	ng3	ng4	ng5	ng6	ng7	class
seq1	1	1	1	0	0	0	1	1
seq2	1	1	1	1	1	0	0	1
seq3	1	1	0	1	1	0	0	1
seq4	1	0	0	0	0	0	0	1
seq5	1	1	1	0	0	0	0	1
seq6	1	1	1	1	1	0	0	1
seq7	1	1	0	1	1	0	0	1
seq8	1	0	0	1	0	0	0	1
seq9	1	0	0	0	0	0	0	1
seq10	0	1	0	1	0	1	0	2
seq11	0	0	0	1	1	1	0	2
seq12	0	1	0	1	0	1	0	2
seq13	0	0	0	1	1	1	0	2

Fig. 4. Boolean data table

- 50% according to the less frequent family. In this case, we just save the n-grams that exist at least in 50% of F2, i.e. those of a frequency>=2. As a result, the set of saved n-grams is {ng1, ng2, ng4, ng5,ng6}.

The main conclusion that we can draw from this example is that ng6 is eliminated by the first two hypotheses. However, it exists in 100% of F2's sequences and in 0% of F1's sequences. It is a very relevant descriptor because it perfectly discriminates the two families. From this observation, we adopt the third approach. At last but not least, we must define the right S_{min}'s value. In fact, this problem is resolved with experiments. The true value depends on the type of the data and the opinion of the domain expert.

3.4 Algorithms and Theoretical Comparison between the Two Approaches

Clssical algorithm

E : n-grams set with a single size
Fich : protein file
M : n-grams set with different sizes
n : n-grams size

Begin
M =NULL
E =NULL
n:=2

do
E := Extract_n_grams (n, NULL, Fich)
M := M + E
n:=n+1
E:=NULL
While(n<=5)
return (M)
End

Hierarchical algorithm

E : n-grams set with a single size
Fich : protein file
M : n-grams set with different sizes
S_{min} : n-grams support
n : n-grams size
Begin
M :=NULL
E :=NULL
S_{min} := constante
n:=2
do
E := Extract_n_grams (n,E,Fich)
E := Filter(E,S_{min})
M := M + E
n :=n+1
While (n<=5)
return (M)
End

Table 1. CPU time(second) for n-grams extraction by the tow approaches(average of 10 files)

	Trivial Approach	Smin=30	Smin=50	Smin=70
Average	318	584	250	155

According to the table 1, we observe the influence of the parameter S_{min}. The computation time decreases when the parameter increases. When we set $S_{min} >= 50$, the computation time is significantly small in regard to the trivial approach, which consists in extracting all the n-grams.

4 Experiments and Results

4.1 Data and Evaluation Methods

Data bank. In order to evaluate the efficiency of this new approach, we used real biological dataset. We randomly draw 5 protein families from the SCOP database [10]. SCOP gathers different types of proteins family structures. The used classification is organised in several hierarchical levels: super families, families and folds. There are approximately 50 observations in each family. Our goal is to discriminate a family in regard to another family by using a supervised learning algorithm. We carry out a pairwise analysis i.e. we want to differentiate each pair of families. Thus, we build 10 datasets based on our 5 families. Each dataset contains about 100 observations.

Error rate evaluation and classifier. In order to estimate the prediction error rate, we use a 10 x 2 cross-validation that we repeat many times. This technique gives a rather good estimation [11]. The second problem is the classifier choice. We select a linear SVM (Support Vector Machine with a linear kernel) as a classifier. In a previous work, we carried out a comparative study of linear and non-linear classifiers such as RBF-SVM (Radial Basis Function kernel), CART Decision Trees, Naive Bayes classifier, nearest neighbor, etc [12]. The main result is that the SVM with a linear kernel is one of the most efficient classifier in our context. It seems that because we combine a restrictive representation bias (linear classifier) and learning bias (maximum margin principle), this classifier is particularly stable, with a very strong resistance to the overfitting. It was the main issue in our context where we combine a very high dimensionality and a small number of available examples.

4.2 Results

In this section, we will present the results in the following way:

1. We describe the results of the standard n-grams approach. We try various value of "n" and detect the best one (Table 2 and 3). It will be our reference in the examination of results, especially in order to evaluate the efficiency of the hierarchical approach.
2. We describe the results of the hierarchical method. We examine the impact of the S_{min} parameter on the number of descriptors obtained (Table 4).

Table 2. Average number of extract n-grams for 10 file of protein families couple

	2-grams	3-grams	4-grams	5-grams	Total (2+3+4+5)
Avg(F_XY)	399	6658	28432	37412	72902

3. Finally, we examine the results according to three ways:
   - We compare the standard 3-grams to the trivial approach where we set together all the 2+3+4+5-grams.
   - We compare these approaches to our hierarchical approach.
   - On the hierarchical approach, we inspect the impact of the S_{min} parameter on the accuracy of the resulting classifier. The underlying question is: is there an "optimal" value that we can infer on all protein families discrimination?

Standard approach: First, we want to mention that we limit the length "n" of n-grams to 5. Beyond this value, our computer is not operational because there are too many descriptors, it is crashed because a lack of memory.

The table 2 describes the number of n-grams (descriptors) for each value of n. These n-grams are built with the classical approach. In the last column, we gathered all n-grams between n=2 and n=5. The first conclusion from the table 2 is that the number of n-grams is large (except 2-grams), it rapidly increases with the n-value. The result is coherent; even if we do not reach the theoretical number of n-grams when n increases, the number of obtained descriptors is still large.

The right value of "n" remains generally an open question. Only the experiments can supply an answer, which is limited to the dataset and the classifier used. In our context, n=3 seems a good approximation. We may be confident to this indication because: we randomly draw the protein families of our study, in this point of view, the result may be extended to the other families; in a previous work where we used a very different classifier (a nearest neighbours classifier), n=2 gives also a valuable results [6].

Hierarchical approach: As mentioned earlier, this approach uses a minimum support to filter the n-grams in the feature extraction process. The parameter enables to master the resulting number of descriptors. If the parameter is too restrictive, we obtain a very small number of descriptors; numerous relevant descriptors may be filtered out.

Table 3. Comp**arison** between extracted n-grams with classical approach

	2-grams	3-grams	4-grams	5-grams
F12	0.0186	0.0198	0.0919	0.2314
F13	0.0511	0.0851	0.1096	0.1500
F14	0.0275	0.0242	0.0417	0.0600
F15	0.0583	0.0500	0.0667	0.1037
F23	0.0260	0.0240	0.0470	0.0970
F24	0.0086	0.0117	0.0367	0.0977
F25	0.0202	0.0193	0.0579	0.1281
F34	0.0590	0.0507	0.0769	0.0940
F35	0.0262	0.0393	0.0779	0.1008
F45	0.0291	0.0324	0.0405	0.0561

If we set a permissive value, the resulting number of descriptors is very large. Numerous irrelevant descriptors will disturb the classification task. The S_{min} parameter setting plays an important role on the computation time and the accuracy of the subsequent classifier.

Let us also note that we limit the maximum length of n to 5 in our experiments. It is not an intrinsic limitation of the approach or of the used computer. We set this limitation in order to obtain comparable results with the other methods of this paper (standard and trivial approaches). In effect, we can also let pursue algorithm without limitation on the n-grams length *(n > 5)* until that one has no more valid n-grams (i.e. frequent). We test various number of S_{min} values (30,50,70).

In table 4 we present the average number of resulting descriptors in our 10 datasets according to the 3 values of tested S_{min}.

Table 4. Evolution of n-grams average number with filtering (S_{min} =30%-50%-70%)

	2-grams		3-grams		4-grams		5-grams		n-grams *(S_{min}filter)*
	Init	S_{min}	Init	S_{min}	Init	S_{min}	Init	S_{min}	
Avg(30%)	399	385	6583	1175	10450	199	580	119	1879
Avg(50%)	399	364	6425	385	3853	53	165	30	833
Avg(70%)	399	352	6272	138	1471	19	63	10	520

We observe several interesting results:

- The resulting number of descriptors of the hierarchical process is reasonable (Table 4), in comparison to the trivial approach.
- In a logical way, the larger is the S_{min} parameter, the smaller is the number of descriptors obtained.
- The larger is the length of the n-gram, the stronger is the filtering effect. We observe this phenomenon in the difference between the columns *Init* and S_{min} in table 4.

4.3 Comparison between the Various Approaches

Note that, E_2 is the set of 2-grams (standard); E_m is the set of a 2+3+4+5-grams without filtering (trivial); E_{m30}, E_{m50} and E_{m70} are the sets of descriptors obtained from the hierarchical approach with respectively S_{min} =30, S_{min} =50, S_{min} =70. Each set has two properties *Nb* and *err_rate* (Table 5), they represent the number of resulting n-grams and the error rate obtained with the classifier C-SVC (linear SVM).

Table 5. Evaluation of n-grams set for the tow approaches

		Nb	Error rate
Standard approach	E_2	399	0.0324
Trivial approach	E_m	71761	0,0423
Hierarchical Approach	E_{m30}	1902	0,0309
	E_{m50}	848	0,0325
	E_{m70}	531	0,0322

E2 against Em: In this part, we compare the error rates of the obtained classifications with the $\mathbf{E_2}$ set, and this which are obtained with the E_m set. $\mathbf{E_2}$ shows a significantly better accuracy as $\boldsymbol{E_2}(err\_rate) << E_m(err\_rate)$; moreover, the number of descriptors is really smaller $\boldsymbol{E_2}(Nb) < E_m(Nb)/10$. Thus, with a smaller number of n-grams, we get better error rates. The standard approach with $n=2$ totally outperforms the trivial approach, in both the computation and the accuracy considerations. However, we are not satisfied with the results because we described in previous sections the drawbacks of being limited to one length of n-grams. The biologists expect descriptors with varying length.

Em against {Em30, Em50, Em70}: The proposed solution is a heuristic solution where we extract the n-grams and filter them in an hierarchical way by respecting the support S_{min} value. We varied the value of S_{min} between 30, 50 and 70. The table 5 shows that with the three values we always find excellent accuracy than the E_m set. Even the results of E_{m30}, E_{m50} and E_{m70} are also better than the $\mathbf{E_2}$ set.

This result is particularly interesting in regard to the composition of the selected set of descriptors obtained. According to table 4, we observe that we have a mix of different descriptors. Using a feature ranking process we could associate the most relevant descriptors to the families in a detailed examination of the results with the biologists.

The results evolution according to $\mathbf{E_{m30}}$, $\mathbf{E_{m50}}$ and $\mathbf{E_{m70}}$: As a selected solution, we varied Smin between three values 30, 50 and 70. We noticed from the table 5 that $E_{m30}(err\_rate)$, $E_{m50}(err\_rate)$ and $E_{m70}(err\_rate)$ are very similar, $E_{m30}(err\_rate)$ is the best one in this experiment. Regarding the number of obtained descriptors, if we compare $E_{m30}(Nb)$, $E_{m50}(Nb)$, and $E_{m70}(Nb)$, we say that E_{m70} is the best one because the selected descriptors is smaller to E_{m30}. The main result is most of all that the process is not very sensitive to this parameter about the accuracy rate. This means we select $\boldsymbol{S_{min}}$ according to the treated problem, the demands of biologists, etc.

5 Conclusion

In this paper, we outline a new descriptor extraction approach in a protein classification context. This approach is named "hierarchical approach" and consists in building n-grams that have variable length to better respond to the biologists' needs. We compared this approach to the standard approach of the n-grams, where we define first the "n" value and later we extracted the n-grams with n-length.

Indeed, we used just one type of the n-grams. The results show that the accuracy of the new approach. The extracted n-grams have good classification rates. Moreover, the resulting descriptors number is reasonable. This enables to carry out the construction of the supervised classifier in good conditions. A further important advantage is that the results are consistent with the biological domain knowledge. The extracted descriptors are compatible with the notion of patterns and the fields of the proteins' family.

We can improve these results. This can be realized by the reducing of the number of n-grams extracted by the hierarchical approach. Indeed, the hierarchical approach operates in a unsupervised way. There are certainly a large number of redundant

descriptors regarding to their relevance in the prediction. Using efficient supervised feature selection process should be removing more descriptors without a deterioration of the accuracy [17].

References

1. Fayyad, U., Shapiro, G., Smyth, P.: From data mining to knowledge discovery: A overview. In: Advances in Knowledge Discovery and Data Mining, pp. 1–34. MIT Press, Cambridge (1996)
2. Gibas, C., Jambeck, P.: Introduction à la bioinformatique, Oreilly (2002)
3. Karplus, K., Barrett, C., Hughey, R.: Hidden Markov models for detecting remote protein homologies. Bioinformatics 14, 846–856 (1998)
4. Falquet, L., Pagni, M., Bucher, P., Hulo, N., Sigrist, C.J.A., Hofmann, K., Bairoch, A.: The PROSITE database, its status in 2002. Nucleic Acids Res. 30, 235–238 (2002)
5. Sebastiani, F.: Machine learning in automated text categorisation. ACM Survey 34(1), 1–47 (2002)
6. Mhamdi, F., Elloumi, M., Rakotomalala, R.: Textmining, features selection and datamining for proteins classification. In: IEEE/ICTTA 2004 (2004)
7. Mhamdi, F., Elloumi, M., Rakotomalala, R.: Desciptors Extraction for Proteins Classification. In: Proceeding of NCEI 2004, New Zealand (2004)
8. Lallich, S., Teytaud, O.: Évaluation et validation de l'intérêt des règles d'association, n°spécial Mesures de qualité pour la fouille des données, Revue des Nouvelles Technologies de l'Information, RNTI-E-1, 193–218 (2004)
9. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proceedings of the 20th VLDB Conference, Santiago, Chile (1994)
10. Murzin, G.A., Brenner, E.S., Hubbard, T., Chothia, C.: SCOP, a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Bio. 247, 536–540 (1995)
11. Dietterich, T.: Approximate statistical tests for comparing supervised classification learning. Neural Computation journal 10(7), 1895–1924 (1999)
12. Rakotomalala, R., Mhamdi, F.: Évaluation des méthodes supervisées pour la discrimination de protéines. In: Dans le proceeding de la conférence SFC 2006, Metz (2006)
13. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and other kernel-base learning methods. Cambridge University Press, Cambridge (2000)
14. Eddy, S., Mitchison, G., Durbin, R.: Maximum discrimination hidden Markov models of sequences consensus. Journal of Computational Biology 2, 9–23 (1995)
15. Krogh, A., Brown, M., Mian, I.S., Sjolander, K., Haussler, D.: Hidden Markov models in computational biology: Applications to protein modeling. Journal of Molecular Biology 235(5), 1501–1531 (1994)
16. Vapnik, V.: The nature of statistical learning theory. Springer, Heidelberg
17. Guyon, I., Gupta, H.: An introduction to variable and feature selection. Journal of Machine Learning Reasearch, 157–1182 (2003)

Extension of Schema Matching Platform ASMADE to Constraints and Mapping Expression

Sana Sellami [1], Aicha-Nabila Benharkat [1], Rami Rifaieh[2], and Youssef Amghar[1]

[1] LIRIS, National Institute of Applied Science of Lyon, Villeurbanne, France
{Sana.Sellami,Nabila.benharkat,Youssef.Amghar}@insa-lyon.fr
[2] San Diego Supercomputer Center, University of California San Diego, San Diego, USA
rrifaieh@sdsc.edu

Abstract. Information systems' activities are increasingly becoming distributed. Many systems, therefore, need to exchange documents in order to correctly perform a critical activity. A growing number of document exchange behavior creates new requirements, such as automatically matching document structure, simplifying user's tasks in creating mappings, and automatically generating data transformation. In our work, we are interested in all these processes: matching, mapping, and data transformation. We propose, in this paper, to extend the XML schema matching used for document exchange to cover constraints management. We show how the constraints improve the performance of schema matching. We also propose XME (XML Mapping Expression) which is an expression model of mappings along with a collection of transformation operators. These extensions take part in our platform for document exchange (ASMADE) that automates matching and generates data transformation instances.

Keywords: Integration, XML schemas, Constraints, Schema Matching, Schema Mapping, Mapping Expression Model.

1 Introduction

The current informational environment is characterized by strongly distributed systems. Reducing back-office costs and enhancing the speed and effectiveness of these systems can be reached with message based data exchange. These data are wrapped in a well formatted exchanged document. For instance, health systems follow HIPAA (Health Insurance Portability and Accountability Act) [19] to offer a simplified way for managing health financial claims and reimbursement through EDI (Electronic Data Interchange) standards.

Data interchange standards specify structure and content of messages or data sets being exchanged between systems [24]. However, data stores on the originating and receiving systems rarely have identical structure. As consequence, data interchange commonly requires that the data be transformed twice: once when the message is assembled at the originating site, and again when it is parsed at the receiving site [21].

In order to optimize the transformation process, schema matching was suggested in [2] [20] for XML based EDI. In this respect, schema matching identifies a set of

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 223–234, 2009.

similarity between the components of two document/data structure. Furthermore, a unified platform called ASMADE (Automated Schema MApping for Documents Exchange) was suggested in [17] for handling matching, mapping and code generation.

We are interested, in this paper, in extending ASMADE platform. Firstly, we suggest a new dimension for schema matching that includes constraints. The constraints are essential source of semantic integrity for exchanged data. In XML based document exchange model, these constraints can be defined using XML Schema constraints on the structure and content of the documents. The constraints in a schema may also include data type assignments that affect how information is processed.

Secondly, we define the process of mapping XML schemas and we propose XME (XML Mapping Expression) which is an expression model for mappings. This model defines theoretically usable transformations rules following the matchmaking generated by the process of matching. We describe over this model a collection of transformation operators to cover useful cases for the generation of target elements.

This paper is organized in the following way. In section 2, we discuss related works concerning matching and mapping processes as well as the constraints managements. In section 3, we propose the extension of EXSMAL (matching and filtering layer) algorithm proposed in [2] to constraints management. In section 4, we describe our proposed XML Mapping Expression (XME). In Section 5, we present all these extensions in ASMADE architecture. Section 6 presents a prototype of this platform. Finally, we conclude the paper with the discussion and future work.

2 Related Works

The automation of the data integration and schema matching has been studied for a long time. We can distinguish between two categories of approaches: matching and mapping.

2.1 Management of Constraints in Schema Matching

The Matching is an operation that takes in input two data schemas and returns the semantic similarities values of their elements. Several works have been achieved in order to provide matchmaking algorithms that manage the correspondences or the incompatibilities of the schema. We are especially interested in matching algorithms that manage constraints. In the literature, Cupid [3] treats the constraints of schemas as the types of data and the ranks of values, the cardinality, etc. Similarity Flooding [4] [5], structural Matching algorithm for XML schemas, SQL DDLS, RDF schemas, UML and OEM, treats the constraints that are based on the primary keys, unique keys or referential constraints that refer to the foreign and constrained keys of cardinality. XClust [6] treats the constraints of cardinality that can be imposed on the elements of the DTD as well as the constraints of similarity, compatibility. In the setting of our approach, we will work more precisely on the EXSMAL [2] algorithm. We search therefore to extend the EXSMAL algorithm to constraint management. We study, therefore, the different constraints related to the XML schemas as described in [22], [23].

2.2 Schema Mapping

In the second category, the process of mapping consists of completing the correspondences values with the true semantic relation. Several models have been proposed for this goal, and offer operations of transformations that are given by the semantic relations such as equivalent, more general, incompatible, compose, is composed of, etc. The LIMXS model [7] (Layered Interoperability Model for XML Schemas) offers a semantic view for XML schemas through the concepts specification and semantic relationship among them. This model is based on a set of transformation operations on schemas' entities that can be composed with each others to represent more transformations. The approach proposed in [8] is an extension of the LIMXS approach called XML Hyperschema. In Model Management System [9], the operators are generic and are applied only on the schemas (called models). The approach presented in [10] [11] creates an interactive Mapping based on correspondences values that show how the value of a target attribute can be created from a set of source attributes values. The mapping expressions are studied like values of correspondences. A Mapping Expression Model has been proposed in [1]. This model is an extension of the approach proposed in [10] since it adds some conditions and defines a filter for source within mapping expression between the similar elements. In addition to all these models, several tools achieving the mapping exist on the market. We have established a comparative study in [12] between these different tools and we have identified the limits of each. We conclude that all these tools don't realize simultaneously the matching, mapping and code generation. Our purpose is to realize a platform that combines all this process to simplify the task of users working on document exchange.

3 Extension the Matching Process to Constraints Management

3.1 Constraints Matching Classification

We consider that the search of the correspondences between the XML schemas elements must benefit from various types of constraints [18]. These constraints are an essential source of semantics for matching between elements. The use of these constraints plays a key role in pruning away weak matches. In other words, the information about elements' constraints can be used as discriminators to determine the likelihood that two elements are equivalent [25]. We distinguished two categories of constraints: the intrinsic constraints and the process constraints.

3.1.1 Intrinsic Constraints

The intrinsic constraints are defined and applied on the elements of the XML Schema. These constraints are generally related to the information on the schemas. This information could be the representation of the elements, their names, their textual descriptions and structures, relations between the elements, the statutes, and the cardinality of the elements. The intrinsic constraints can be classified in two categories [13] covering simple and complex type in XML schemas.

C_1: Constraints on the simple types
The simple types don't contain other elements or attributes. These simple types are defined or are predefined. Their constraints are defined by the facets as the minimum, maximum value for the numbers, dates and lengths, and the patterns, the minimum, maximum length for the chains, names, ID.

```
Example: <xsd:simpleType>
<xsd:restriction base="xsd:positiveInteger">
<xsd:maxExclusive value="100"/>
</xsd:restriction>
</xsd:simpleType>
```

C2: Constraints on the complex types
Complex types are the elements that contain other elements and/or attributes. Their constraints include:

· *Ct: Type Constraint:* These are supplementary constraints data while associating some datatypes to the XML document data. These constraints concern predefined datatypes (i.e, primitive and derived datatypes).

```
Example:<xsd:restriction base="xsd:string">
          <xsd:pattern value="[^0-9]"/> </xsd:restriction>
```

· *Cp: Property Constraint:* An XSD attribute element can have three optional attributes: use, default and fixed. These attributes translate primary keys of which the values are not null and unique. Combinations of these three parameters define what is acceptable in the XML final document (e.g. mandatory attribute, optional, default value, etc.).

```
Example: <xsd:attribute name="partNum" marks="SKU" uses="required"/>
```

· *Co: Occurrence Constraint:* the occurrence is determined by the attributes minOccurs that can only be a positive integer and not null and maxOccurs that can also be a positive integer and not null or unbounded. These attributes are controlled by the attribute "use" such as required, optional or prohibited values.

```
Example:<xsd:element name="item" minOccurs="0" maxOccurs="unbounded">
```

· *Cc: Composition Constraint:* These constraints are the constraints on the elements (i.e.: sequence, choice, all).

· *Cs: Structural Constraint:* These constraints correspond to the determination of the structure of the document XML. The structure defines especially the XML elements: name, number and the nature of their attributes as well as their content.

```
Example: <xsd:element name="element_name" type="xsd:type_name"/>
```

· *Cr: Reference Constraint [14]:* These constraints can be a source of semantics for the relations between the elements and more precisely between the attributes. This type of information can be determined through the ID attributes and IDREF.

· *Cn: Conceptual Constraint:* If two elements are matched they must be similar to the same concept. This concept is defined by user predefined ontologies.

· *Cg: Global Constraint:* These are the constraints about the elements and the global attributes that are referenced by "ref".

3.1.2 Process Constraints

Process constraints are specified by the user or by the experts of the domain. These constraints can be added or modified according to the needs. They play a key role in the determination of semantic correspondences between the schemas in the domain and are important because they are essential part of consistency framework. If these constraints are observed during the process of matching, we can potentially improve

the accurateness of the matching. Thus, we add a filter to our architecture that eliminates the matching that violates these constraints.

· *Cf: The Frequency Constraint:* imposes some regularities to which a source schema must be compliant. For example at most one element source matches CITY or precisely an element source matches STATE.

· *Ca: The Adjacent Constraints:* imply that if two elements are not related as "street and quantity" meaning that they cannot participate in the process of the matching. The user, in this case, can use this constraint for never generate a candidate to the matching that could combine street and quantity.

· *Cm: The Mutual Constraint* [16]*:* expresses for example one element e belonging to a schemas source S can matches an element targets f which belongs to a targets schema T if and only if e is one of the elements most similar to f and mutually if f is one of the elements most similar to e.

· *Cu: The Uniqueness Constraint:* verifies that every element must be matched with a distinct element in the schemas targets.

·*Cmp: The Constraint of previous matchings* [15]*:* takes into consideration a previous matching result, and permits to exploit it to minimize the matching operations.

3.2 Matching Intrinsic Constraints

We propose an algorithm that treats the constraints on the simple types (C_1), type (Ct) and occurrence (Co) constraints.

3.2.1 Algorithm Description

Let EC_1 be a set of matched elements. We are going to determine for every node e_i of the source schema S and node f_j of the target schema T if their correspondences don't violate the intrinsic constraints. Here we wish to determine the matching which respect the constraints and filter EC_1 pairs with respect to the penalty violation of the constraints.

A brief description of the used methods is as follows:

IsCompatible: this method returns the compatibility between the constraints; it covers all the possible cases of simple and complex constraints. The compatibility is true if e_i and f_j have the same type of constraints. The different cases are when: e_i *has* simple with f_j *has* simple constraints or e_i *has* simple with f_j *has* simple and complex constraints (we ignore the complex and just match the simple); e_i *has* complex with f_j *has* complex constraints; e_i *has* complex with f_j *has* simple and complex constraints (ignore the simple) and finally e_i *has* simple and complex with f_j *has* simple and complex. **isConstraint**(e_i, C_k)***:*** returns true if C_k is an existent constraint for (e_i). **isFiniteNumericConstraint**: covers all the constraints that are specified on the numeric finite domain like C_1, Ct, Co. ***ConstraintType:*** This method determines the type of constraint defined on the element. It can be simple or complex type. ***ComputePenaltyViolation:*** This method calculates the penalty of violation to the matching constraints. This value is calculated in function of the weight of the constraint and the value of the violation of constraint with respect to NbCommonConstraint. ***ComputeConstraintViolation:*** This method calculates the violation of a constraint, we illustrate in the algorithm the method used for numeric finite domain. It uses domain intersection and the cardinality of the intersection versus the cardinality of each domain.

3.2.2 Pseudo-Code

Inputs:

- EC_1: Set of Similarities between nodes (result of matching), $EC_1=\{(e_i ,f_j,val_1)/i=1,n,j=1,m \}$ where $e_i \in$ S (source element) and $f_j \in$ T (target element).
- Constraints C_1 and C_2 on elements: C_1, Cst is the set of constraints on simple types and C_2 is the set of constraints on complex types. Thus, $C_1=(Cst,\omega_0)$ and $C_2=\{(Ct, \omega_1),(Cp, \omega_2),(Co, \omega_3),(Cc, \omega_4),(Cs, \omega_5),(Cr, \omega_6),(Cn, \omega_7),(Cg, \omega_8)\}$ where $\omega=\{ \omega_0, \ldots,\omega_8 \}$ the weight associated with each constraint.

Outputs:

The set of refined correspondences EC_2, where $EC_2= \{(e_i, f_j,val_2)/i=1,n,j=1,m \}$

```
ConstraintMatching (EC1, C1, C2)
Begin
Cpvs(i,j)=Cpvc(i,j)= Cv(i,j)=0
EC2= ∅
For each (ei, fj,val1) ∈ EC1
  If (isCompatible(ConstraintType(ei), ConstraintType(fj)))
 // see isCompatible section 3.2.1
    Then   If (isSimple(ConstraintType(ei)))
             Then Cpvs(i,j)= ComputePenaltyViolation(ei,fj, C1)
             End If
           If (isComplex(ConstraintType(ei)))
              Then Cpvc(i,j)=  ComputePenaltyViolation(ei,fj, C2)
             End If
  End If
  If (Cpvs(i,j) =0 or  Cpvc(i,j)=0)
    Then Cm(i,j)= NotZero Between(Cpvs(i,j),  Cpvc(i,j) )
         // if both 0, it returns 0  //Cm:Constraint Matching
    Else Cm(i,j)= Avg(Cpvs(i,j), Cpvc(i,j))
  End If
  If (Cm(i,j) ≤ 0.5)
     Then     val2=val1
              EC2= EC2 ∪ (ei, fj,val2)
  End If
End For
Return EC2
End


ComputePenaltyViolation(ei, fj , C)
Begin
Int NbCommonConstraint=0
Cpv(i,j)=0 // penalty violation
For each constraint Ck ∈ C
 If (isConstraint (ei, Ck) and isConstraint (fj, Ck))
   Then
    Vc(i,j)= ComputeConstraintViolation(ei, fj, Ck)
    NbCommonConstraint++
    Cpv= Cpv + ωk* (1- Vc(i,j))
// ωk is the associated weight for the constraint Ck
  End If
End For
If (NbCommonConstraint ≠ 0)
   Then   Cpv(i,j)= Cpv(i,j) / NbCommonConstraint
End If
Return Cpv(i,j)
End
```

```
ComputeConstraintViolation(e_i, f_j, C_k)
Begin
 If (isFiniteNumericConstraint(C_k))
  Then  //look for domain compatibility and intersection
   If (D(e_i, C_k) ∩ D(f_j, C_k) ≠ ∅)
     Then If (D((e_i, C_k) ∩ D(f_j, C_k)= D(e_i, C_k) or
                 D((ei, C_k) ∩ D(f_j, C_k)= D(f_j, C_k))
               Then Vc_(i,j)=0
             Else
         Vc_(i,j)=Card(DomainIntersection(e_i, f_j, C_k) )/
           (Card(D(e_i, C_k))+ Card(D(f_j, C_k)))
         End If
    End If
  Else // Other Methods for non numeric constraints
      // not described in this algorithm
End If
Return Vc_(i,j)
End
```

4 XME: XML Mapping Expression

4.1 Mathematic Representation

Based one the Mapping Expression Model defined in [1], we propose XME (XML Mapping Expression Model) that identifies how to generate the value of a target element/attribute from sources values. We define an attribute as follows:

$$A_i = \sum_{i}^{-} [\, f_i(a_r, a_p, ..., a_e), l_i, c_i \,] + \tau_i = \alpha_i(A) + \tau_i \qquad (1)$$

A_i represents an element/field of the target representation. $\alpha_i(A)$ is a mapping expression applied on multi-sources to generate a part of the target field A or to generate a sub-field of the field *A*, $\alpha_i = < f_i , l_i , c_i >$ where f_i is a mapping function. l_i is a set of filters for sources rows, we could find a filter $l_i(a_r)$ for each source field a_r . C_i is a condition on the mapped value of the field A. This enables us to ignore mapped results that do not satisfy the condition. τi represents a generic function, independent of the source. The generic function covers all the cases that the mapping expression model doesn't generate. This function can be a constant value or, an incremental function, an automatic function or a skolem function. We present (in fig.1) the XME model (expressed in XSD) that defines all the properties of the XML schemas as well as the used transformation operators.

4.2 Transformation Operations

The set of primitive transformation operations are the building blocks that would enable schemas transformation. These primitive operations can be composed together to represent larger transformations. They are summarized as follows:

Add: describes an element that appears in the schemas targets but it is not in the source schema.
Remove: describes an element in the schemas source that doesn't appear in the target schema.

Concat: describes an element of the target field that is established by concatenation of the elements of the source. (intra schema).
Split: This is the reverse operation of Concat.
Connect: Corresponds to a Matching 1-1 that joins two elements equivalent without any modification (identity).
Apply: This operator is used when we need some functions to transform the content of the source values into targets values. These functions can be by example the conversion of units or a mathematical function (min, avg, div, etc.).
Join: plays the role of an inter schema concat .
Rename: relates identical elements except that the name of the element or type that change.
Delete: to suppress the elements that doesn't participate in the Mapping.

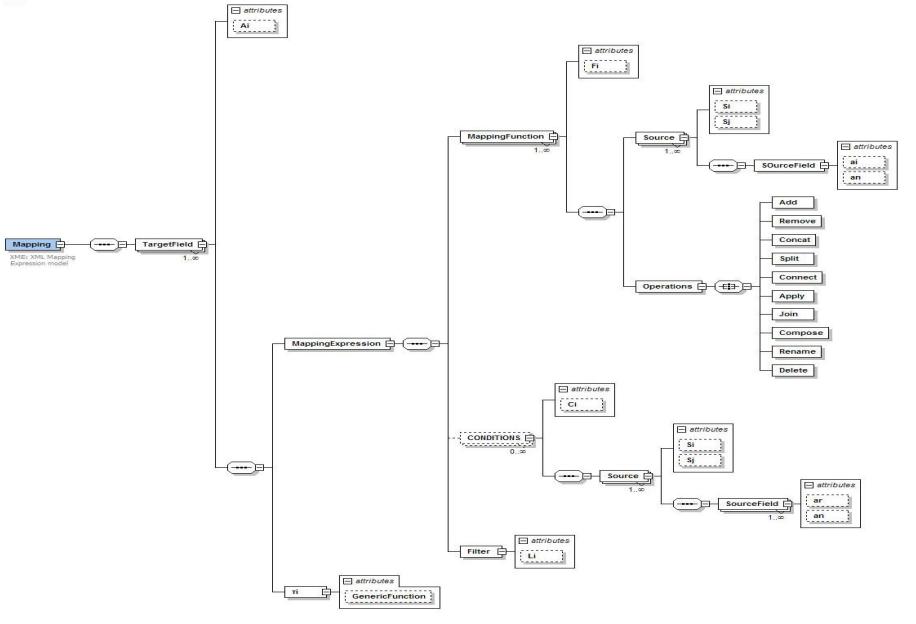


Fig. 1. Representation of XME

5 ASMADE Platform

We present in this section the ASMADE system (Automated Schemas MApping for Documents Exchange) and all the extensions described above (fig.2). This platform was proposed in [17]. ASMADE contains 4 main layers:

- **Matching Layer:** Entries (XSD schemas) are transformed into trees by the EXSMAL algorithm (matching layer). The process of Matching produces EC_1 a set of correspondences (1-1, 1-n, n-m). This set is verified according to the constraints expressed on the elements. EXSMAL has been proposed as a solution for

the EDI message's schema matching [2]. The criteria for matching include data-type, structure, and elements descriptions. The choice for the XML schema was motivated by its potential to define the structure and the semantics of an EDI message. However, the matching in EXSMAL doesn't take yet into account the information related to an element (constraints, status, cardinality). For this purpose, we have proposed and defined a classification of different constraints and an algorithm to take into consideration the intrinsic constraints.

- **Filtering Layer:** A filtering layer permits to eliminate irrelevant correspondences. Herein, we applied matching constraints (intrinsic and process constraints) to prune insignificant matching in EC_1 (set of correspondences).
- **Mapping Layer:** In this third layer, we use the previous result designated by EC_2 to generate some usable rules of transformation expressed in our proposed model XME (XML Mapping Expression).
- **Layer transformation:** In this last layer, the resulting mappings will be generated in XQuery and executed by XQuery engine. Each Mapping will be transformed according to XQuery's FLWR expression (For-Let-Where-Return).

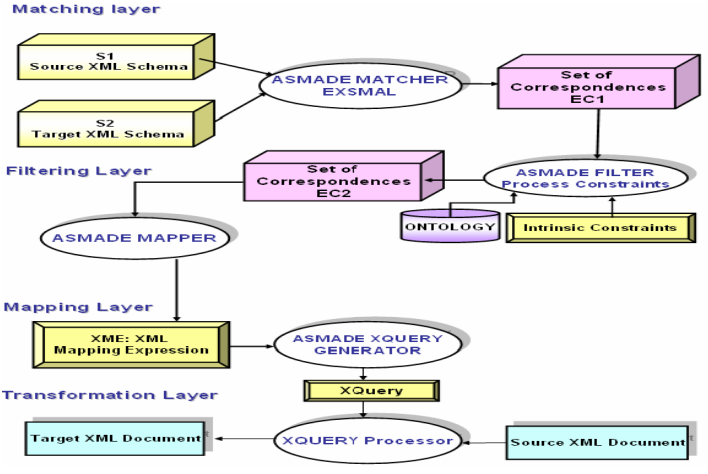


Fig. 2. Extension of the ASMADE Architecture

6 Implementation

We describe in this section an example about using the different transformation operators and an ASMADE prototype.

6.1 Example

We describe in this example how the different operators can be used. In (fig.3), we use the operator Join between $ISBN_{S1}$ (in the s source schema S_1) and $ISBN_{S2}$ (in the source schema S_2) for generating mappings with the element of target schema $ISBN_{TS}$.

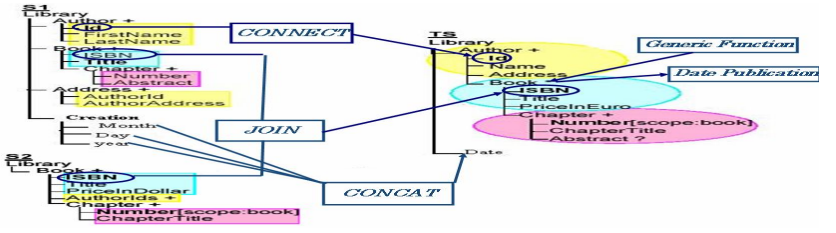


Fig. 3. Example of generated mappings between XML Schemas

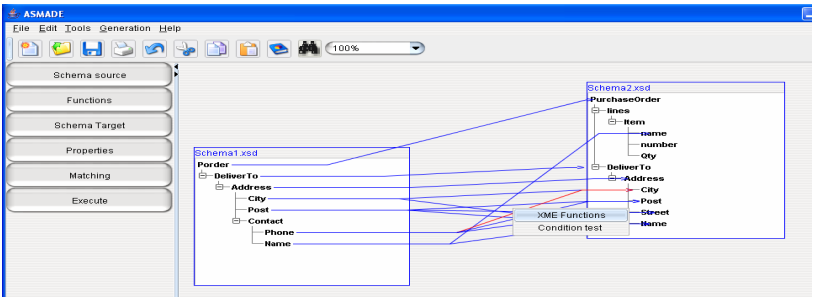


Fig. 4. Snapshot of the matching generated by EXSMAL

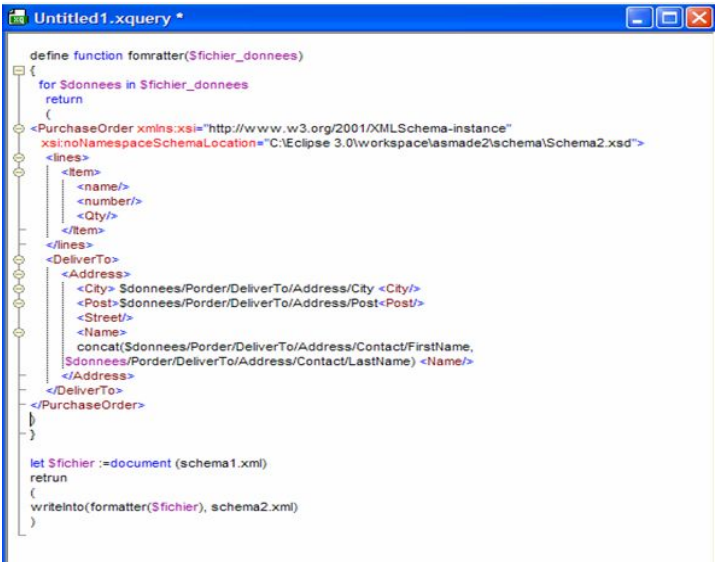


Fig. 5. Snapshot of the resulting mappings generated in XQuery

7 Conclusion and Future Work

Due to the extensive use of XML in several domains as universal data exchange format, there has been a great interest on proposing rich matching and mapping processes for simplifying user's tasks. We have realized a study on the different solutions and approaches proposed to realize the management of constraints in schema Matching as well as the mapping. After that, we have defined the different constraints that can be applied on the elements (sources and destinations) of schemas and established a classification of these constraints. We have also proposed an algorithm which takes into consideration the intrinsic constraints and returns the penalty of violation of constraints to determine the degree of similarity between the schemas. We have proposed also XME (XML Mapping Expression) which is an expression model of mappings and we described a collection of transformation operators to realize the mapping and to determine the semantic relation between schemas. All these propositions appear in the ASMADE architecture (Automated Schema MApping for Documents Exchange) that offers a framework to cope with complete approach of documents exchange. We intend in the future to improve the algorithm of constraints including all the constraints (intrinsic and process), improve the prototype of ASMADE, and integrate the algorithm of constraints. We intend also to extend the matching approach to deal with the large scale schema/ontology matching. The front end of this framework is also indented to be used for ontologies and taxonomies matching in other fields such as data management in biology.

References

1. Rifaieh, R.: Using Contextual Ontologies for Semantic Sharing Within Enterprise Information Systems. Phd Thesis, National Institute of Applied Sciences of Lyon, 202 (2004)
2. Rifaieh, R., Chukmol, U., Benharkat, N.: EXSMAL: EDI/XML semi-automatic Schema Matching Algorithm. In: CEC 2005 Conference, München, Germany, pp. 422–425 (2005)
3. Madhavan, Bernstein, P.A., Rahm, E.: Generic Schema Matching with Cupid. In: VLDB Conference, Rome, Italy, pp. 49–58 (2001)
4. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity Flooding: A versatile Graph Matching approaches. In: ICDE Conferences, San Jose, California, USA, pp. 117–128 (2002)
5. Melnik, S., Rahm, E., Berstein, P.A.: Rondo: A programming Platform for Generic Model Management. In: SIGMOD Conference, San Diego, California, USA, pp. 193–204 (2003)
6. Lee, M., Yang, L.H., Hsu, W., Yang, X.: XClust: Clustering XML Schemas for Effective Integration. In: CIKM Conference, McLean, VA, USA, pp. 292–299 (2002)
7. Boukottaya, A., Vanoirbeek, C., Paganelli, F., Abou Khaled, O.: Automating XML Transformations: A conceptual modelling based approach. In: APCCM 2004, Dunedin, New Zealand, pp. 81–90 (2004)
8. Zerdazi, A., Lamolle, M.: HyperSchema XML: Un modèle d'intégration par enrichissement sémantique de schémas XML. In: IUT Montreal (2005)
9. Melnik, S., Bernstein, P.A., Halevy, A., Rahm, E.: Supporting Executable Mappings in Model Management. In: SIGMOD Conference, Baltimore, Maryland, USA, pp. 167–178 (2005)
10. Miller, J.R., Haas, L.M., Hernandez, M.A.: Schema Mapping as Query Discovery. In: VLDB Conference, Cairo, Egypt, pp. 77–88 (2000)

11. Haas, L.M., Hernandez, M.A., Ho, H., Popa, L., Roth, M.: Clio Grows Up: From Research Prototype to Industrial Tool. In: SIGMOD Conference, Baltimore, Maryland, pp. 805–810 (2005)
12. Sellami, S.: Conception et réalisation d'un outil de génération automatique de mappage pour la transformation de documents XML. National Institute of Applied Sciences of Lyon (2006)
13. Rahm, E., Hong-Hai, D., Maβmann, S.: Matching Large XML Schemas. In: SIGMOD Record, pp. 26–31 (2004)
14. Zamboulis, L.: Technical Report XML Schema Matching and XML Data Migration and Integration: A Step Towards The Semantic Web Vision (2003)
15. Doan, A.: Learning to Map Between Structured Representation of Data. In: WWW, pp. 662–673 (2002)
16. Madhavan, J., Bernstein, P.A., Doan, A., Halevy, A.: Corpus-based Schema Matching. In: ICDE Conference, Tokyo, Japan, pp. 57–68 (2005)
17. Benharkat, N., Rifaieh, R., Herzi, K., Amghar, Y.: ASMADE: Automated Schema Mapping for Documents Exchange. In: SEDE Conference Los Angeles, California, pp. 79–85 (2006)
18. Smiljanic, M., Van Keulen, M., Jonker, W.: Formalizing the XML Schema Matching Problem as a Constraint Optimization Problem. In: Andersen, K.V., Debenham, J., Wagner, R. (eds.) DEXA 2005. LNCS, vol. 3588, pp. 333–342. Springer, Heidelberg (2005)
19. HIPAA, http://www.hipaa.org/
20. Rifaieh, R., Chukmol, U., Benharkat, N.A.: A matching algorithm for electronic data interchange. In: Bussler, C.J., Shan, M.-C. (eds.) TES 2005. LNCS, vol. 3811, pp. 34–47. Springer, Heidelberg (2006)
21. ACM Queue Jouranl. 4(7), 25 (2006), http://www.acm.org/
22. http://www.w3.org/TR/xmlschema-1/
23. Buneman, P., Fan, W., Simeon, J., Welnstein, S.: Constraints for Semistructured Data and XML. In: SIGMOD Record, pp. 45–47 (2001)
24. Rifaieh, R., Benharkat, N.: A Framework for EDI Message Translation. In: AICSSA 2003 Conference, Tunis, Tunisia (2003)
25. Shen, W., Li, X., Doan, A.: Constraint-Based Entity Matching. In: AAAI 2005 Conference, Pittsburgh, Pennsylvania, pp. 862–867 (2005)

An Ontology Based Method for Normalisation of Multidimensional Terminology

Ahlem Nabli, Jamel Feki, and Faîez Gargouri

MIRACL Laboratory
Département d'Informatique, Faculté des Sciences Economiques et de Gestion de Sfax
Route de l'Aérodrome km 4, B.P. 1088, 3018 Sfax, Tunisia
{ahlem.nabli,jamel.feki,faiez.gargouri}@fsegs.rnu.tn

Abstract. The data warehouse design raises several problems such as the integration of heterogeneous data sources. In fact, the main difficulty is how interpret automatically the semantic of the heterogeneous and autonomous data. In addition, in information system design, the use of ontology becomes more and more promising; it contributes to avoid semantic and structural ambiguities. This research introduces the concept of decisional ontology as an assistance tool for the specification of analytical requirements. For this, we propose an ontology based method to standardize the multidimensional terminology extracted from several heterogeneous data sources.

Keywords: Multidimensional concept, decisional ontology, semantic relation, normalisation.

1 Introduction

Nowadays the Data Warehouse (DW) has emerged as a powerful technology that has a growing interest for the scientific community. Indeed, DW represents one of the most significant applications in decisional information system (DS) and database domains. Although the databases and DW have in common the processing of huge volumes of data, their missions are completely different, as far as their vocations. Consequently, the DW design requires techniques completely different from those adopted for operational systems.

The majority of research works in DW focuses on specific subjects such as multidimensional modeling [1], materialized views [5] [16], index selection [6] and data mining. The great attention was devoted to these practical subjects which mainly determine the performances of the DW, while other conceptual issues, like the requirement specification and DW schema design, still need more investigations. Indeed, few works have been made to automate the DW design process [3] [7].

The majority of DW research projects bypass the design phase to concentrate on specific tasks as query optimization or multidimensional modeling. This negligence can be due to the difficulties in specifying and defining decisional requirements. Mainly, we note the absence of graphic tools to assist decision makers to express their requirements. In addition, this requirement expression does not necessitate only

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 235–246, 2009.

knowing what information we need to aggregate, but also depends on how DS are organized (multidimensional structure). Other problems induced by the heterogeneity of the sources emerge; they are known as semantic and structural ambiguities [4][11] such as synonymy in names of facts, dimensions, etc.

Actually, research in ontology domain presents a set of solutions that models Knowledge Engineering [2]. This ontology aims at establishing the representations through which the computer can handle the information semantic. The ontology application fields cover many systems like computer aided decision-making, computer aided educational, software design, etc. More precisely, ontology gives a great satisfaction to solve the semantic and structural conflicts between the various data sources in data base integrating process [4] [11].

Recently, many methodologies were proposed to support the development of ontologies and their use in integration systems [4] [10] [13]. The approach we recommend allows the capitalization of the ontology's advantages in terms of semantic and structural data integration [13]. Thus, we proposed an architecture for decisional ontology construction [18]. This architecture allows ontology building in an incremental and progressive way. It is based on four phases (c.f. Fig 1): *i) extraction* of multidimensional concepts from heterogeneous data sources, *ii) comparison* in order to deduce semantic relations, *iii) upgrade* the ontology with concepts and relations extracted in i) and, finally *iv) optimization* of the ontology relations.

This paper deals with the comparison and upgrade steps to standardize multidimensional terminology concepts. It introduces in Section 2 the usage context and the construction steps of decisional ontology. Section 3 presents the basic elements of decisional ontology in term of multidimensional concepts and outlines extraction heuristics. Section 4 details comparison and upgrade steps. Section 5 synthesizes the proposal and presents our future works.

2 Decisional Ontology

A decisional ontology is a representation of knowledge dedicated to the decisional systems. It can be defined as a referential of multidimensional concepts of a field, and their semantic and multidimensional relations [18]. Its use covers different steps of the DW life cycle, since the requirement specification, via DW and Data Mart schema design until the evolution of DW. During these steps, a decisional ontology assists the designer to solve problems of data sources heterogeneity. In the OLAP requirement specification step, the decisional ontology helps to validate the multidimensional concepts (fact, measure, dimension....) and relations between these concepts. For example, it is useful to notify "*illegal*" associations between two concepts such as the association of a dimension called *supplier* to a fact analyzing *Sales*. Naturally, decisional ontology prevents associations between concepts not semantically associable (e.g. associating fact-to-fact, dimension-to-dimension, hierarchy-to-fact, etc).

In [9] [14] and [19] the construction method of an ontology is based on a sequence of three steps: the *extraction*, *comparison* and *upgrade*. For a decisional ontology we define the four following steps: *i) extraction*, *ii) comparison*, *iii) optimization* and, *iv)*

upgrade. This adaptation gave rise to an optimization step. Figure 1 depicts these steps detailed below.

Extraction. It is the first step (Fig 1-①) of the decisional ontology construction. It extracts multidimensional elements from a given data source. This step is manual and builds the initial version of the decisional ontology. It is subdivided into three sub-steps i) *extraction of the Multidimensional Concepts* C_{MD} (fact, measures, dimension, etc.), ii) *confirmation* of C_{MD} extracted from the previous stage by the designer and, iii) *extraction of multidimensional relations* (i.e. fact-measure relation, fact-dimension relation, etc) between these concepts.

Comparison. The second step (Fig 1-②) of a decisional ontology construction consists of the semantic comparison of each C_{MD}, extracted from the previous step, with the ontology content. This comparison is based on a set of semantic relations and uses a set of rules to deduce the adequate relation between two compatible C_{MD} (i.e. of the same type). This stage allows solving syntactic and semantic ambiguities between multidimensional concepts. It is followed by a standardization of these C_{MD}.

Upgrade. The upgrade (Fig 1-③) of the decisional ontology begins by the insertion of the C_{MD} and their deduced semantic relations, then it continues by the insertion of their multidimensional relations.

Optimization. Once the semantic relationships between C_{MD} are entered into the ontology, an optimization phase (Fig 1-④) is necessary. It is to discover the impact of the inserted semantic relations deduced between C_{MD} and the existing ones in the ontology. This optimization uses a set of inference rules.

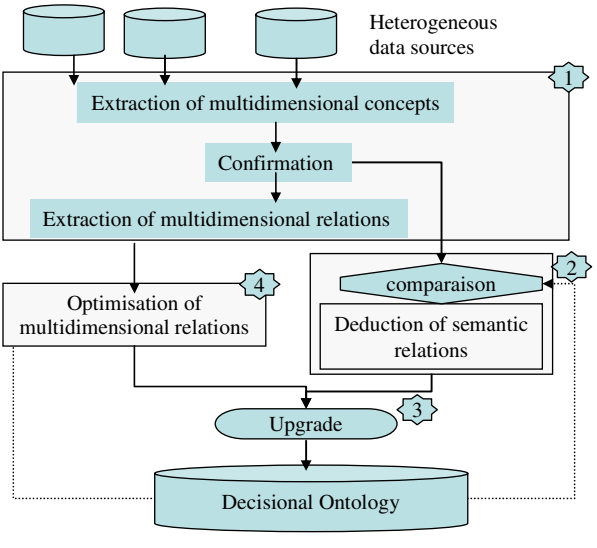


Fig. 1. Architecture for a system of construction of decisional ontology

Our method of decisional ontology construction is progressive and iterative. The assistance of the designer throughout this construction is necessary or even mandatory. It allows the approval of the results obtained and helps to solve ambiguities.

As decisional ontology has specific concepts and in order to improve this paper readability, we first define its concepts and briefly describe our extraction rules of multidimensional elements. Finally, we detail our method for multidimensional concepts standardization.

3 Extraction of Multidimensional Concepts

In order to understand the extraction and the standardization heuristics, we start by presenting the definitions of the basic concepts of multidimensional modeling then, we outline the extraction heuristics detailed in [18].

3.1 Multidimensional Concepts

Fact Concept. In decisional systems, the analyzed data are modeled as subjects of analysis called *Fact*. Any subject of analysis is then represented by the *fact Concept (F)*.

Measure Concept. Each fact is characterized by one or more attributes generally numerical; these attributes constitute indicators of analysis and are represented by the *Concept of measure (M)*.

Dimension Concept. The perspectives (axes) according to which measures are recorded and analyzed, are represented by *the Concept of dimension (D)*.

Parameter and weak attribute Concepts. A dimension is made up of attributes [12]. In a dimension some attributes take part to define levels of detail (hierarchy), they are known as strong attributes or parameters and are represented by *the concept of parameter (P)*; other attributes do not define levels of detail but can however be useful to label results for example; they are qualified as *weak attributes* and are represented by *the concept of weak attributes (Wa)*.

Hierarchy Concept. The parameters of a dimension *D* are organized in hierarchies ordered from the finest to the highest granularity. Each hierarchy is a directed acyclic graph of some or all parameters of *D*. It represents *the concept of hierarchy (H)*.

3.2 Extraction Heuristics

The multidimensional concepts are extracted using a set of extraction rules. These rules are based on the entities, relationships between entities of the conceptual model describing the data source, the relationship cardinalities and, on the type of attributes which they contain [18].

Fact identification: Each entity of the data source verifying one of the two following heuristics is identified as a fact.

An n-ary relationship in the data source having numerical attributes with $n \geq 2$. (F1)

An entity having at least one numerical attribute not included in its identifier. (F2)

Measure identification: since measures are numerical attributes, they will be searched within identified facts (heuristic F1 or F2) and in "*parallel*" entities. Heuristics are the following:

Every non-key numerical attribute of a fact *F* is a measure for *F*. (M1)

Every non-key numerical attribute of a parallel entity to fact F is a measure for *F*. (M2)

Every numerical attribute of entities related to *F* by a "one-to-one" link first, followed by those related to F by a "one-to-many" link is a measure for *F*. (M3)

Dimension attribute identification: Each attribute, not identified as a measure and verifying the following two rules becomes a dimension attribute.

An attribute in a entity identified as a fact *F*. (D1)

An attribute of an entity related directly or indirectly to a PF via a "one-to-one" or "one-to-many" link. (D2)

The output of this step is a set of multidimensional concepts extracted from different data sources. Table 1 represents an sample set of multidimensional concepts extracted from two databases for the commercial domain. To standardize them we first compare these concepts using ontology to deduce the semantic relations then we group them and assign a generic name for the group.

4 Standardisation of Multidimensional Concepts

As decisional ontology is a knowledge representation dedicated to decisional systems, it is defined as a referential of multidimensional concepts for a specific field, their semantic and multidimensional relations.

This standardization requires a comparison of the extracted concepts to determine the semantic relations, and a mapping of these concepts.

More precisely, to standardize the OLAP concepts, we first compare the names of these concepts to determine the semantic relations [15]; this may use an existing ontology, then factorizing them. The factorisation is based on the determined relations to assure the mapping between the concepts.

4.1 Comparison of Multidimensional Concepts

It is a stage devoted to:

- identify the similar occurrences of concepts,
- specify link between them and,
- detect possible conflicts between these occurrences of concepts.

This comparison requires the study of the conflicts between the names of these concepts by using a thesaurus containing semantic relations illustrated by names and set of relations between these names, such as synonymy, homonymy etc. This thesaurus is specific to a domain and maintains for each term its synonyms, aliases … recorded in a structured way.

To determine the semantic correspondence between the multidimensional concepts, we defined the following criteria adapted from [15]:

- Fact name (measure name) comparison criteria to compare semantically the fact names (resp. measure names);
- Dimension comparison criteria to compare semantically the names of hierarchies and there parameters;

These criteria express the semantic relationship among multidimensional concepts gathered from different data sources. We define five types of relations between concepts:

- *Synonymy(C1,…,Cn)*: the synonymy relationship between *n* concepts *C1,…,Cn* expresses that these concepts converge on the same meaning.

Synonymy(returned, returns, revenue)

- *Equivalence(C1,…,Cn)*: different concepts *C1,…,Cn* issued from different data sources may converge on the same meaning.

Equivalence(Returned,income)

- *Identity(C1,…,Cn)*: the identity relationship between *C1,…,Cn* issued from different data sources expresses that these concepts have the same name and meaning.
- *Homonymy(C1,…,Cn)*: the same concept can have two different meanings. We use the term of homonymy to indicate the state, where there is not any semantic relationship between two identical forms. This relationship is represented by *homonymy*.
- *Antonymy(C1,C2)*: denotes the absence of any implication relationship between the two concepts. On the other hand, the negation of the one implies the assertion of the other; we cannot deny both at the same time.

Note that the previous semantic relations are applicable on facts, measures, parameters, dimensions.

The comparison phase produces a set of semantic relationships between multidimensional concepts.

4.2 Mapping of Multidimensional Concepts

After the determination of the semantic relations between multidimensional concept occurrences, we proceed to their mapping by applying the following heuristics.

This step concerns the name of facts and their measures and, the dimension names and their parameters.

The mapping consists in grouping several names of concepts having the same type and relations of equivalence, identity or synonymy, and then assigning them a significant name. For that, we define a function named *Significant* (p_1, p_2…, p_n) where:

- p_1, p_2…, p_n are names of concepts of the same types (i.g. all are facts).
- The returned result is the most significant name among p_1, p_2…, p_n.

If the semantic relation between two fact names is a homonymy relation, we define a new name for one of them.

Construction and mapping of fact names

For any fact concept, we distinguish the two flowing cases:

If there is a set of facts $F_1,\ldots,F_N$ such as *Synonymy (Fi…,Fj)*$\wedge$ *Equivalence(Fk…,Fl)* $\wedge$*Identity(Fh…,Fr)* where i,j,k,l,h,r $\in$ [1,N] , then these facts describe a same fact. All

these name facts are grouped and are represented with a unique name using the function *significant*; this name is added to the ontology as a standard (generic) name of the *N* fact names of this group.

If there is two fact occurrences F_i *and* F_j such as *Homonymy*(F_i, F_j) (i.e. F_i *and* F_j describe two different facts with the same name) we change the name of one of them and add both of them to the ontology.

Construction and mapping of measure names. The standardization of measure names proceeds in a similar manner to the standardization of fact names.

For any measure concept, we distinguish the two flowing cases:
Given *SM* a set of measures $M_1,\ldots,M_N$, if *Synonymy* $(M_i,\ldots,M_j) \wedge$ *Equivalence*$(M_k,\ldots,M_l) \wedge$ *Identity*$(M_h,\ldots,M_r)$ where $i, j, k, l, h, r \in [1,N]$ then this *SM* describes the same measure, thus we represent measures of *SM* by a unique name returned by the function *significant*. The returned name is added to the ontology as a standard name of this group *SM*.

If there is two measures M_i *and* M_j such as *Homonymy*(M_i, M_j) *then* we represent them in the ontology by two different names assigned by the designer.

Example: Figure 2(a) presents two measures named respectively *quantity* and *qty* which are related with the synonymy semantic relation. Consequently, these two measures describe a single measure. Then, the function *significant* is called significant (*quantity*, *qty*) returns "*quantity*" and the returned name is added to the ontology as the generic name for the two measures; Figure 2(b) illustrates this example.

```
<Concept_M >
    <measure  RWE="DB1" name ="quantity" />
    <measure  RWE="DB2" name ="qty" />
    <semantic_rel>
        <synonymyies>                                          (a)
            <synonymy>
                <name_s RWE="DB1">quantity</name_s>
                <name_s RWE="DB2">qty</name_s>
            </synonymy>
            ……
<Concept_M>
    <semantic_rel>
        <synonymies>
            <synonymy Generic_n="quantity">                    (b)
                <name_s RWE="BD1">quantity</name_s>
                <name_s RWE="BD2">qty</name_s>
            </synonymy>
        </synonymies>
     ……..
```

Fig. 2. Extract of the decisional ontology before standardization (a) and after standardization (b) of measures

Construction and mapping of dimensions. To standardize dimensions, we compare their names and their parameters. The standardization of dimension names proceeds in a similar manner to the standardization of fact names.

Here the standardization begins by the parameters correspondence, then hierarchies' correspondence in order to decide for the dimension standardization.

In the following we denote:

-*d* is a dimension,
-*P* is a parameter
-*P(d)* means that *P* is a parameter of *d*.
-*h* is a hierarchy
-*h(d)*: means that *h* is a hierarchy of d,
-*P1<P2*: means that *P1* has a lower granularity than *P2*.
-*h=[P1<P2<…<Pn]* : *P1, P2, .., Pn* are parameters of *h* organized from the finest to the highest granularity.

Mapping parameters.

For any parameter concept, we distinguish the two flowing cases:

If there is a set of parameters $P_1(d1)..$, *Pn(dc)*, such as *Synonymy(Pi(da)..,Pj(ds)) ∧ Equivalence (Pk(de)…,Pl(df)) ∧Identity(Ph(dv) …,Pr(du))* where i,j,k,l,h,r ∈[1,n] and a,s,e,f,v,u ∈[1,c], then these parameters describes a same attribute, thus we represent them with an unique name given by the function *significant*; then we add it to the ontology as a standard name for this group.

If there is two parameters *Pi, Pj* such as *Homonymy(Pi,Pj) and Pi=Pj*, then *Pi, Pj* describe different attributes with the same name, thus we represent them with different names assigned by the designer then they are added to the ontology.

Example: let us consider an extract of the decisional ontology presented in Fig 3(a). It contains a set of parameters and their semantic relation. For the year parameter we have the identity semantic relation (*Identity(year(*$date_{DB1}$*),year(*$date_{DB2}$*))* see Fig 3(a.1)) and the synonymy relation between *year* and *yr* in the same database (*sunonymy(year(*$date_{DB1}$*),yr(*$date_{DB2}$*))* see Fig 3(a.2)). Consequently, these three parameters describe a same parameter. Then, the function *significant* is called "*significant* (*year*, *year,yr*) = *year*" and the returned name is added to the ontology as the generic name of the presented parameters as illustrated in Figure 3(b.1). For the homonymy relation identified with the address parameter from the two data bases (see Fig 3(a.3)), we change the name of one of them as illustrated in Fig 3(b.2).

Mapping hierarchies. Once the correspondences between the parameter names are made, we continue by the mapping of hierarchies. In fact, we use the results of the parameter standardisation to compare hierarchies. Several cases can arise:

Case 1: Identity of hierarchies. The identity of parameters occurs when parameter names represent a relation of equivalence, identity or synonymy.

Let us *hx* and *hy* two hierarchies of the same length with *hx*=[*Px1<Px2<…<Pxn*] and *hy*=[*Py1<Py2<…<Pyn*]

```
<Concept_P>
    <param name ="address" source="BD1"/>
    <param name ="yr" source="BD1" />
    <param name ="year" source="BD1" />
    <param name ="year" source="BD2" />
    <param name ="family" source="BD2" />
    <param name ="address" source="BD2" />
        ……..
    <semantic_rel>
        <identities>
            <identity>
                <name RWE="BD1">year</name>
                <name RWE="BD2">year</name>                    (1)
            </identity>
            ……                                                              (a)
        </identities>
        <synonymyies>
            <synonymy>
                <name RWE="BD1">year</name>
                <name RWE="BD2">yr</name>                      (2)
            </synonymy>
            ……
        </synonymyies>
        <homonymies>
            <homonymy>
                <name RWE="BD1">address</name>
                <name RWE="BD2">address</name>                 (3)
            </ homonymy >
        </ homonymies>
            ……..
        </semantic_rel>
</Concept_p>


<Concept_P>
 <param name ="function" source="BD1" />
        <semantic_rel>
            <identities>
                <identity Generic_p="year">
                    <name RWE="BD1">year</name>                (1)
                    <name RWE="BD2">year</name>
                    <name RWE="BD2">yr</name>
                </identity>
                …….                                                         (b)
            </identities>
            <homonymies>
                <homonymy exist_name="address">
                    <name RWE="BD1">address</name>             (2)
                    <name RWE="BD2">delivery_addr </name>
                </homonymy>
            </homonymies>
…….
        </semantic_rel>
 </Concept_p>
```

Fig. 3. Extract of the decisional ontology before standardization (a) and after standardization (b) of parameters

These two hierarchies are identical if and only if $\forall\ i \in [1,n] \wedge$ *significant* (*Pxi, Pyi*). Then *significant* (*hx*,*hy*) can be called to return the their generic name.

Consequently, we represent the hierarchies with a unique name obtained by the function *significant*. This name is added to the ontology as a standard name of *hx* and *hy*.

Case 2: Inclusion of hierarchies. This case happens when the parameters of hierarchy *hx* are included in hierarchy *hy* of the same dimension.

Let us *hx* and *hy* two hierarchies with *hx*=[*Px1*<*Px2*<…<*Pxn*], *hy*=[*Py1*<*Py2*<…<*Pym*] and n<m

The hierarchy *hx* is included in *hy* if and only if $\forall\ i \in [1,n] \wedge \exists\ j \in [1,m] \wedge$*significant* (*Pxi, Pyj*). In this case *hy* is generic then the function *significant*(*hx*, *hy*) returns *hy*. The name of *hy* is added to the ontology as the significant hierarchy.

Case 3: Intersection of hierarchies. Certain parameters of hierarchy are included in other hierarchy. That means that, the hierarchies have attributes in intersection. In this case we have two sub cases: *i*) all their parameters represent one single hierarchy or, *ii*) they represent different hierarchies of the same dimension. The designer should intervene to decide to merge the two hierarchies or not.

Let us *hx* and *hy* be two hierarchies with *hx* = [Px1<Px2<…<Pxn] and *hy* = [Py1<Py2<…<Pym]

These two hierarchies are in intersection if and only if $\exists\ i \in [1,n] \wedge \exists\ j \in [1,m]$ $\wedge$*significant (Pxi,Pyj)*.

The designer can merge *hx* and *hy* in a same hierarchy (which is added to the ontology) or keep the two separated hierarchies.

Case 4**:** Disjoint hierarchies. This case occurs when we do not find any semantic relation between parameters of hierarchies of the same dimension. Indeed, the hierarchies have disjointed parameters.

Let us *hx* and *hy* two hierarchies with *hx*=[Px1<Px2<…<Pxn] and *hy*=[Py1<Py2<…<Pym]

These two hierarchies are disjoint if and only if $\forall\ i \in [1,n] \wedge \forall\ j \in [1,m] \wedge$ *significant* (*Pxi,Pyj*)=ϕ then *hx* and *hy* are two different hierarchies.

In this case, the dimension has two hierarchies. If these hierarchies have the same name then we identify the homonymy semantic relation between them. We must change one of them. Then, they are added to the ontology.

Example: let us consider the hierarchies for the *Client* dimension extracted from two databases DB1 and DB2 as presented in Fig 4(a)(b). Based on the parameter standardization we do not find any semantic relation between parameters of hierarchy DB2.h1_client and parameters of hierarchies DB1.h1_client and DB1.h2_client. Then, we have disjointed hierarchies between h1_client from DB2 and all of h1_client and h2_client from DB1. In this case we change the name of the hierarchy DB2.h1_client and we add it to the ontology as the third hierarchy for the *Client* dimension (cf. Fig 4(c)).

Case 5**:** Distinct hierarchies. Distinction of hierarchies occurs when several parameter names have the *Antonymy* relation.

```
<Concept_D>      (a)
    <dimension name ="client" source="DB1" identif_d="id_client"/>
    <dimension name ="client" source="BD2" identif_d="id_client"/>
    ……..
                    (b)
<hierarchies>
    <hierar_dim name_d="client">
    <hierarchy source="DB1">
        <name_h> h1_client </name_h>
        <name_h> h2_client </name_h>
        </hierarchy>
   <hierarchy source="DB2">
        <name_h> h1_client </name_h>
    </hierarchy>
    <semantic_rel>
    <homonymies>
        < homonymy>
        <name_s RWE="BD1">h2_client</name_s>
        <name_s RWE="BD2">h1_client</name_s>
        </ homonymy >
    </ homonymies >
```

```
                    (c)
<hierarchies>
  <hierar_dim name_d="client">
  <hierarchy>
     <name_h RWE="DB1"> h1_client</name_h>
     <name_s RWE="BD1">h2_client</name_s>
     <name_s RWE="BD2">h3_client</name_s>
  </hierarchy>
  < homonymies >
   < homonymy exist_n="h1_client">
     <name_s RWE="BD1">h1_client</name_s>
     <name_s RWE="BD2">h3_client</name_s>
   </ homonymy >
   </ homonymies >
```

Fig. 4. Extract of the decisional ontology before standardization (a,b) and after standardization (c) of hierarchies for the *Client* dimension

Let us *hx* and *hy* two hierarchies with *hx*=[Px1<Px2<…<Pxn] and *hy*=[*Py1<Py2<…<Pym*]

These two hierarchies are distinct if and only if $\forall$ i $\in$ [1,n] $\wedge$ $\forall$ j $\in$ [1,m] $\wedge$ *significant* (*Pxi, Pyj*)= ϕ $\wedge$ $\exists$ *Antonymy(*Pxi, Pyj*)*. They are added to the ontology as hierarchies of different dimensions.

Dimension construction. The dimension construction is based on the result of the previous step. We distinguish two situations as follows.

Situation 1: for the first four previous cases the hierarchies describe a single dimension, thus we represent these dimensions as a single one with a unique name obtained by the function *significant*. This dimension name is added to the ontology as a standard dimension name.

Situation 2: for the fifth case the hierarchies describe different dimensions, thus we represent each one separately. Then, each dimension and its hierarchy are added to the ontology as different dimensions.

The result of this step is a set of multidimensional terms standardized and represented in the decisional ontology.

5 Conclusion and Future Works

This paper first outlines our method of decisional ontology construction. It then detailed the standardisation of the ontology multidimensional terms. The proposed method starts from multidimensional concepts extracted from heterogeneous data sources. Then, these concepts are standardized and loaded to the decisional ontology.

We expect three application fields for the decisional ontology. The first examines how to automate the standardization of multidimensional concepts. The second consists in integrating the decisional ontology to allow requirement specifications. Finally, and as a long term research axis how the evolution of the decisional ontology may be done.

References

1. Agrawal, R., Gupta, A., Sarawagi, S.: Modeling Multidimensional Databases. Research Report, IBM Almaden Research Center. In: ICDE 1997, San Jose, California (1995)
2. Benslimane, D., Arara, A., Yetongnon, K., Gargouri, F., Ben Abdallah, H.: Two approaches for ontologies building:From-scratch and From existing data sources. In: The 2003 International Conference on Information Systems and Engineering ISE 2003, Montreal, Canada, July 20-24 (2003)
3. Bonifati, A., Cattaneo, F., Ceri, S., Fuggetta, A., Paraboschi, S.: Designing Data Marts for Data Warehouses. ACM Transactions on Softw. Engineering Methodology (2001)
4. Bouzeghoub, M., Farias Lóscio, B., Kedad, Z., Soukane, A.: Heterogeneous data source integration and evolution (Extended abstract). In: Hameurlain, A., Cicchetti, R., Traunmüller, R. (eds.) DEXA 2002. LNCS, vol. 2453, p. 751. Springer, Heidelberg (2002)
5. Cabibbo, L., Torlone, R.: A logical Approach to Multidimensional Databases. In: Proc. of the 6th Int'l Conference on Extended Database Technology, Valencia, Spain, pp. 187–197 (1998)
6. Firestone, J.M.: Dimensional Object Modeling. Executive Information Systems, Inc., White Paper n. 7 (April 1998)
7. Golfarelli, M., Rizzi, S.: Designing the Data Warehouse : Key Steps and Crucial Issues. Journal of Computer Science and Information Management 2(3), 1–14 (2001)
8. Golfarelli, M., Rizzi, S.: Methodological framework for Data Warehouse Design. In: DOLAP 1998, USA (1998)
9. Gruber, T.R.: Toward Principles for the Design of Ontologies, Used for Knowledge Sharing. Stanford Knowledge Systems Laboratory (1993)
10. Guarino, N. (ed.): Formal Ontology and Information Systems. IOS press, Amsterdam (1998)
11. Haddar, N., Gargouri, F., Ben Hamadou, A.: Model Integration: The Comparison Step. In: Proc. International Conference on the UK Academy for Information Systems, Leeds Metropolitan University, Leeds UK (April 2002)
12. Kimball, R.: The Data warehouse Toolkit. John Wiley & Son, Chichester (1996)
13. Maiz, N., Boussaid, O., Bentayeb, F.: Un système de médiation basé sur les ontologies pour l'entreposagevirtuel. In: INFORSID 2006, Hammamet-Tunisie, May 31-June 3 (2006)
14. Mhiri, M., Mtibaa, A., Gargouri, F.: Towards an approach for building information systems'ontologies. In: FOMI 2005, 1st workshop Formal Ontologies Meet Industry, Verona, Italy, June 9-10 (2005)
15. Mhiri, M., Gargouri, F., Benslimane, D.: Détermination automatique des relations sémantiques entre les concepts d'une ontologie. In: INFORSID 2006, Hammamet-Tunisie, May 31-June 3, pp. 263–271 (2006)
16. Moody, D., Kortnik, M.: From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design. In: DMDW 2000, Sweden (2000)
17. Nabli, A., Soussi, A., Feki, J., BenAbdallah, H., Gargouri, F.: Towards an Automatic Data Mart Design. In: ICEIS 2005, May 3-6. IEEE, Los Alamitos (2005)
18. Nabli, A., Feki, J., Mhiri, M., Gargouri, F.: Vers une approche de construction d'ontologie décisionnelle: Extraction des Eléments Multidimensionnels. In: MCSEAI 2006, Decembre 6-9, Maroc (2006)
19. Noy, N., Mcguiness, D.: Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Medical Informatics Report, SMI-2001-0880 (2001)

Semantic and Conceptual Context-Aware Information Retrieval

Bénédicte Le Grand[1], Marie-Aude Aufaure[2,3], and Michel Soto[1]

[1] Laboratoire d'Informatique de Paris 6 – 8, rue du Capitaine Scott – F-75015 Paris
{Benedicte.Le-Grand,Michel.Soto}@lip6.fr
[2] Supélec – Computer Science department – plateau du Moulon – 3, rue Joliot Curie – F-91192 Gif sur Yvette Cedex
Marie-Aude.Aufaure@supelec.fr
[3] Axis Research Team – INRIA – Domaine de Voluceau – BP 105 – F-78 153 Le Chesnay

Abstract. This paper presents an information retrieval methodology which uses Formal Concept Analysis in conjunction with semantics to provide contextual answers to Web queries The *conceptual* context defined in this article can be *global* - i.e. stable- or *instantaneous*- i.e. bounded by the global context. Our methodology consists first in a pre-treatment providing the global conceptual context and then in an online contextual processing of users' requests, associated to an instantaneous context. Our information retrieval process is illustrated through experimentation results in the tourism domain. One interest of our approach is to perform a more relevant and refined information retrieval, closer to the users' expectation.

Keywords: Context-dependant semantics, emergent semantics in information retrieval systems, Formal Concept Analysis.

1 Introduction

This paper presents a context-aware semantic information retrieval tool. Our goal is to use **conceptual analysis** in conjunction with **semantics** in order to provide **contextual answers** to users' queries on the Web.

In this paper, we present our methodology and show experimentation results of an information retrieval performed on selected tourism Web sites. The information retrieval process is divided into two steps:

- Offline pre-treatment of Web pages;
- Online contextual processing of users' requests.

The pre-treatment consists in computing a conceptual lattice from tourism Web pages in order to build an overall *conceptual context*; this notion is defined in sections 2 and 3. Each concept of the lattice corresponds to a cluster of Web pages with common properties. A matching is performed between the terms describing each page and a thesaurus about tourism, in order to label each concept in a standardized way.

Whereas the processing of tourism Web pages is achieved offline, the information retrieval is performed in real-time: users formulate their query with terms from the

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 247–258, 2009.

thesaurus. This cluster of terms is then compared to the concepts' labels and the best-matching concepts are returned. Users may then navigate within the lattice by generalizing or on the contrary by refining their query.

This method has several advantages:

- Results are provided according to both the context of the query and the context of available data. For example, only query refinements corresponding to existing tourism pages are proposed;
- The added semantics can be chosen depending on the target user(s);
- More powerful semantics can be used, in particular ontologies. This allows enhanced query formulation and provides more relevant results.

This paper is organised as follows: section 2 introduces the notion of *context*, in the general sense and in the field of computer science. Section 3 briefly describes Formal Concept Analysis and Galois Lattices, and defines our global and instantaneous *conceptual contexts*. Our methodology for a semantic coordination of conceptual contexts and ontologies –or thesauri- is proposed in section 4. Finally, we conclude and give some perspectives of this work.

2 Notion of Context

Context is an abstract notion which cannot be precisely defined because it only makes sense when it is linked to a particular situation. Human beings implicitly associate a context to a set of actions, an attitude, etc. in situations of everyday life: context surrounds and gives meaning to something else. Some definitions of context have emerged in cognitive psychology, philosophy and areas of computer science like natural language processing.

The concept of formal context was introduced by McCarthy [1] [2]. According to Giunchiglia, who has also worked on context formalization, "a context is a theory of the world which encodes an individual's subjective perspective about it". This theory is *partial* –incomplete- and *approximate* as the world is never described in full detail [3].

Context is a key issue for many research communities like artificial intelligence, mobile computing, problem solving, etc [4] [5]. In artificial intelligence, means to interact between contexts are defined by rules allowing navigation from one context to others [6]. Contexts can be represented by conceptual graphs, topic maps, description logics with OWL extensions, etc.

As for the Semantic Web, context is often used either as a filter for disambiguation in information retrieval [7], to define contextual web services [8] or as a means to integrate or merge different ontologies [9] [10]. Context could be specified with different granularity levels (document, web page, etc.). Additional information, i.e. the *context*, could then be linked to each resource.

3 Conceptual Contexts and Relationship with Ontologies

In the previous section, we have presented various definitions of *context*. In this article, we define *conceptual contexts*, based on Formal Concept Analysis and Galois

lattices in particular. Many research works apply concept lattices to information retrieval [11]. Formal concepts can be seen as relevant documents for a given query. The introduction of domain ontology, combined with concept lattices to enhance information retrieval is more recent. In [12], the authors propose an approach based on Formal Concept Analysis to classify and search relevant data sources for a given query. This work is applied to bioinformatics data. A concept lattice is built according to the metadata associated to the data sources. Then, a concept built from a given query is merged in this concept lattice. In this approach, query refinement is performed using domain ontology. The refinement process of *OntoRefiner*, dedicated to Semantic Web Portals [13], is based on the use of domain ontology to build a Galois Lattice for the query refinement process. The domain ontology avoids building the whole lattice. Finally, the *CREDO* system [14] allows the user to query web documents and to see the results in a browsable concept lattice (http://credo.fub.it). This system is useful for quickly retrieving the items with the intended meaning, and for highlighting the documents' content.

In [7], the authors investigate methods for automatically relaxing over-constrained queries based on domain knowledge and user preferences. Their framework combines query refinement and relaxation in order to provide a personalized access to heterogeneous RDF data. Contrarily to this approach, our method is dedicated to imprecise and user-centered queries.

In our proposition, Galois lattices are built in order to represent the web pages' content. The user can then browse the lattice in order to refine or generalize his/her query. Compared to the approaches described above, our proposition is not only dedicated to information retrieval, but can also be used for other purposes like populating ontologies, comparing web sites trough their lattices, helping web site designers, etc.

This section is organized as follows: after a short introduction to Galois lattices, we propose our definition of global and instantaneous conceptual contexts.

3.1 Introduction to Formal Concept Analysis and Galois Lattices

FCA is a mathematical approach to data analysis which provides information with structure. FCA may be used for conceptual clustering as shown in [15] and [16].

The notion of Galois lattice for a relationship between two sets is the basis of a set of conceptual classification methods. This notion was introduced by [17] and [18]. Galois lattices consist in grouping objects into classes that materialise concepts of the domain under study. Individual objects are discriminated according to the properties they have in common. This algorithm is very powerful as it performs semantic classification. The algorithm we implemented is based on the one that was proposed in [19].

Let us first introduce Galois lattices basic concepts.

Let two finite sets E and E' (E consists of a set of objects and E' is the set of these objects' properties), and a binary relation $R \subseteq E \times E'$ between these two sets. Figure 1 shows an example of binary relation between two sets. According to Wille's terminology [20], the triple (E, E', R) is a formal context which corresponds to a unique Galois lattice, representing natural groupings of E and E' elements.

Let P(E) be a partition of E and P(E') a partition of E'. Each element of the lattice is a couple, also called concept, noted (X, X'). A concept is composed of two sets X ∈ P(E) and X' ∈ P(E') which satisfy the two following properties :

$$X' = f(X) \text{ where } f(X) = \{ x' \in E' \mid \forall x \in X, xRx' \}$$
$$X = f'(X') \text{ where } f'(X') = \{ x \in E \mid \forall x' \in X', xRx' \} \quad (1)$$

A partial order on concepts is defined as follows:

Let C1=(X1, X'1) and C2=(X2, X'2),

$$C1 < C2 \Leftrightarrow X'1 \subseteq X'2 \Leftrightarrow X2 \subseteq X1 \quad (2)$$

This partial order is used to draw a graph called a Hasse diagram (named after Helmut Hasse (1898–1979)), as shown on the left-hand side of figure 1. There is an edge between two concepts C1 and C2 if C1<C2 and there is no other element C3 in the lattice such as C1<C3<C2. In a Hasse diagram, the edge direction is upwards. This graph can be interpreted as a representation of the generalisation / specialisation relationship between couples, where C1<C2 means that C1 is more general than C2 (and C1 is above C2 in the diagram).

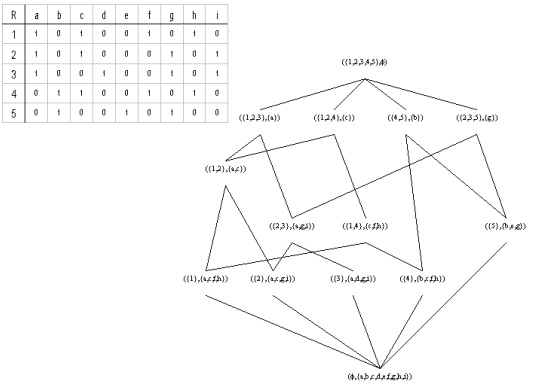


Fig. 1. Binary relationship and associated Galois lattice representation (Hasse diagram)

The concept lattice shows the commonalities between the concepts of the context. The first part of a concept is the set of objects. It is called *extension*. The second set – the *intention* - reveals the common properties of the extension's objects. The right-hand side of the figure 1 shows the Hasse diagram of a concept lattice.

3.2 Definition of Conceptual Context

In our definition of *conceptual context*, we distinguish the *global* conceptual context and the *instantaneous* conceptual context:

- The *global* conceptual context is the result of step 1 of our proposed methodology (see section 3 for details), i.e. a Galois Lattice constructed on a selection of tourism web sites where each concept is a set of web pages labelled using terms of a thesaurus –or an ontology.

Properties:

- The main property of this global conceptual context is to be stable. The only way to make a global conceptual context evolve is to either change the set of selected tourism web sites or/and to change the set of selected properties for a web page as illustrated on figure 2.
- A given web page may belong to different global conceptual contexts. If we consider a Web page about the *Louvre* museum, its global conceptual context is different according to the Web site this page is extracted from (e.g. Web site dedicated to tourism, to national museums, personal Web site, etc.).The similarity with other web pages will depend on this global conceptual context.

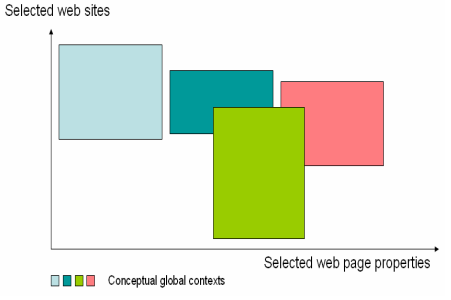


Fig. 2. Variation of global conceptual contexts

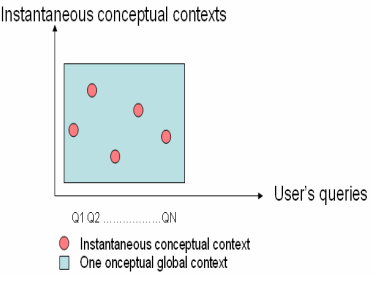


Fig. 3. Variation of instantaneous conceptual contexts

- The *instantaneous* conceptual context is the result of step 2 of our proposed methodology (see section 3 for details) i.e. both the query formulated by the user and this query's result.
  Properties:
  - Because of the explorative and iterative features of the information retrieval process, the main property of this instantaneous conceptual context is that it evolves every time the user modifies, refines or generalizes his/her query.
  - The variation of the instantaneous conceptual context is bounded by the global conceptual context as illustrated in figure 3.
  - A given Web page may belong to different instantaneous conceptual contexts thanks to the generalisation / specialisation relationship resulting from the Galois lattice.

The instantaneous context can be seen as the *current concept* of the lattice, i.e. it indicates the position of the user in his navigation process. The user's initial instantaneous context corresponds to the result of a mapping query. From this starting point, the user may navigate within the global context (the lattice): his/her instantaneous context changes each time he/she travels to another concept through the generalization or specialization links of the global context; the instantaneous context may thus be seen as a vehicle which enables the user to go from one global context to another.

When two global contexts are overlapping, a direct navigation from one to the other is possible. Nevertheless, it is possible to reach isolated global contexts via a semantic layer consisting of thesauri (or ontologies) which have connections with the lattices.

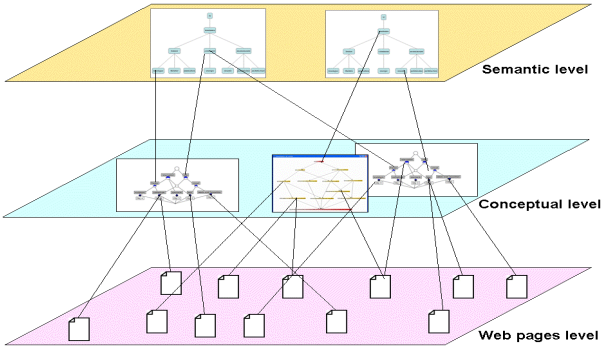


Fig. 4. Three navigation levels

The figure 4 illustrates the three levels of navigation: the lowest level corresponds the resources (that is, Web pages), whereas the second level represents the conceptual layer consisting of the global contexts (the Galois lattices). Finally, the upper level gathers thesauri and ontologies which provide alternative ways of travelling from one global context to another (through the instantaneous context). The instantaneous context may thus be used to navigate both at the conceptual level and between the conceptual and semantic levels.

The information provided by the conceptual context (global or instantaneous) is complementary to intrinsic information about the web page (properties in our case, i.e. most significant terms in the page).

4 Our Methodology for a Semantic Coordination of Conceptual Contexts and Ontologies

Our semantic and context-aware information retrieval methodology can be divided into two steps: offline pre-treatment and online contextual processing of users' queries. In this section, we describe these operations and we illustrate them with a simple example in the tourism domain.

4.1 Step 1: Offline Pre-treatment

4.1.1 Selection of Tourism Web Sites

The first step consists in building a global conceptual context for a given domain by selecting a set of relevant Web pages. These pages may belong to a single or to several Web sites. For illustration purposes, we selected 5 Web pages from the Web site of a French city's tourist office (Metz, in the North-East of France):

http://tourisme.mairie-metz.fr/. This set of Web pages constitutes the information base within which the semantic and context-aware information retrieval is achieved.

4.1.2 Web Pages Parsing: Generation of Input for the Galois Lattice (Objects and Properties)

The global conceptual context is the Galois lattice built from the selected Web pages; we thus need to generate appropriate input for the computation of the conceptual lattice:

- Each Web page corresponds to an *object;*
- The *properties* of a Web page are the most frequent and significant nouns of the page (extracted with our tool based on Tree-Tagger, as illustrated on figure 5).

The list of objects and their corresponding properties is stored in a mysql database, as shown on the figure 5. For example, the *object 3* (corresponding to a specific Web page) is described by the properties *spectacle (show)* and *reservation*.

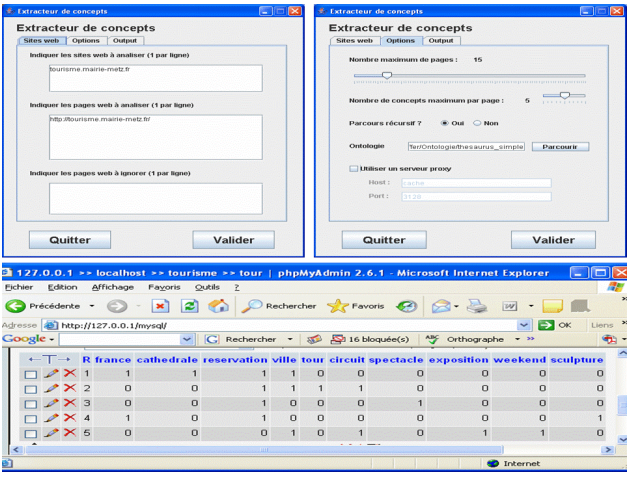


Fig. 5. Objects/properties extraction and the generated database

4.1.3 Construction of a Galois Lattice

The database described above –containing objects and properties- is used as an input for the computation of a Galois lattice.

The output is a lattice of concepts where each concept consists in a set of objects which have some properties in common. The list of objects of a concept is called the concept's *extension* and the corresponding shared properties constitute the concept's *intention*. The lattice generated from the database of figure 5 is illustrated on figure 6.

The lattice shown on figure 6 contains 12 nodes (concepts), among which an upper and a lower bound. Each concept is characterized by its extension –the list of objects it contains- and by its intention –the list of common properties of the extension's objects. The node situated at the top of the lattice is the more general as it contains all

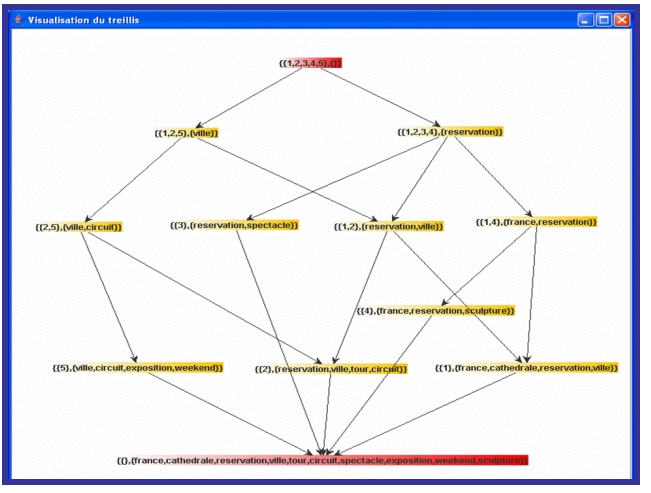


Fig. 6. Global conceptual context: Galois lattice

objects, which have no property in common. The more we go down in the lattice and the more specific the nodes become. For interpretation purposes, it is interesting to study whether and to what extent objects are similar with one another.

This lattice shows us that *object 3* appears in only two concepts because it has only two properties and it has only one property in common with the *objects 1, 2* and *4*: *reservation (booking)*. No object shares the property *spectacle (show)* with *object 3*. On the contrary, *object 2* appears in five concepts because it has more properties (i.e. four) than *object 3* and because its properties are shared by more objects (i.e. four) than *object 3*.
In this case, this means that:

- The global conceptual context of tourism for the French city of Metz is less directed towards shows and more towards the points of interest of the city itself.
- User's queries about shows cannot be refined but only generalized to the concept of reservation and thus are strongly bounded by the global conceptual context.

In other words, if a user is more interested in tourism based on shows, the city of Metz is not the best place –according to the information available on the Web site. We can learn from the user' queries if he/she is really more interested in tourism based on shows than in other kind of tourism. In this case, we can switch to another global conceptual context more "show-oriented". This new context can be linked to the global conceptual context of tourism for the French city of Metz thanks to ontologies as proposed in our conclusion.

4.1.4 Concepts' Labelling

The last step of the offline treatment consists in labelling the concepts of the lattice in a normalized way; with this end, we used the World Tourism Organization (WTO) Thesaurus on Tourism and Leisure Activities. The same operation may be performed with an ontology instead of a thesaurus.

The Web pages' parsing aims at extracting the most frequent nouns of the pages (cf. section 4.1.2). Our analysis goes further, as our tool also matches nouns found in pages with the entries of the WTO thesaurus –through a syntactic matching. For example, the property *spectacle (show)* is linked to the *loisirs (leisure)* entry of the WTO thesaurus. The normalized label of a concept consists in the normalized labels of all the objects of its extension.

This link between raw data –tourism Web pages- and a semantic structure –the WTO thesaurus- aims at making the global conceptual context richer: this context reflects information from original data as well as from general domain knowledge.

4.2 Step 2: Online Contextual Processing of Users' Requests

Once the global conceptual context is built, the instantaneous conceptual context is computed online for each user's query.

4.2.1 Formulation of a Query with Keywords

Users formulate their query with keywords (we may restrict possible keywords to the entries of the WTO thesaurus). The interest of the normalized labels of objects and concepts is that they allow us to use a controlled vocabulary.

4.2.2 Identification of the Best Matching Concept(s) of the Lattice

The answer to a query is the concept in the lattice whose properties contained in the intension best match the query's keywords. If no concept provides a perfect match, the more relevant concepts are proposed to the user through a refinement or a generalization of the query according to available data (in the tourism Web pages), i.e. according to the context of data.

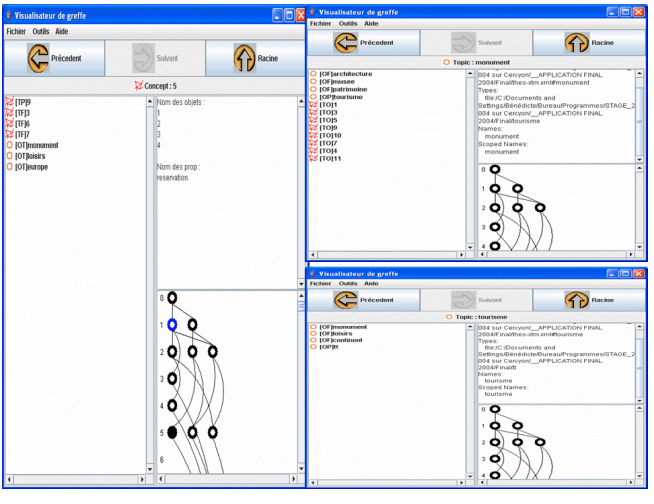


Fig. 7. Navigation within the global context enriched extended with semantic information

The figure 7 shows another visual representation of the Galois lattice, but this interface is richer than the one presented in figure 6 as the links with semantic data –entries of the WTO thesaurus- also appear on the display. Suppose the user entered a query with the keyword *reservation*. The instantaneous conceptual context of this query is illustrated on the screenshot on the left: there is a focus on the concept of the lattice (concept number *5*), whose extension is {*page1, page2, page3, page4*} and the common property of these objects is {*reservation*}. We can also notice that this concept is labelled with three concepts of the thesaurus: *monument, loisirs (leisure), Europe*.

From this instantaneous context (number *5*), the user is free to navigate either within other nodes of the lattice –and thus go to more specific or more general concepts- or within the thesaurus.

The top right corner of the figure 7 focuses on the *monument* entry of the thesaurus, and the left part of the screenshot shows the hierarchy of the thesaurus –*monument* is a subclass of tourism and a superclass of *architecture, museum* and *patrimony*- as well as the other concepts of the lattice which are labelled with the *monument* entry –concepts number *1, 3, 5, 9, 10, 7, 4* and *11*. Once again, the navigation can continue either within the thesaurus or back to the conceptual lattice.

The advantage of this navigation interface is that it allows users to navigate easily within the global contextual context or within the thesaurus –or ontology in general- and to go from one to another transparently. Enlarging the context of data with domain knowledge such as thesaurus or ontology provides richer and more relevant answers to users' queries.

5 Conclusion and Future Work

In this paper, we presented a context-aware information retrieval methodology which uses conceptual analysis in conjunction with semantics. One interest is to perform a more relevant and refined information retrieval, closer to the users' expectation. Our methodology is divided into two steps; the first one consists in an offline pretreatment of web pages where a conceptual lattice is built from tourism Web pages. Each concept corresponds to a cluster of Web pages having common properties. Then, a matching is performed between the Web pages' relevant terms and a thesaurus about tourism, in order to label each concept in a normalized way. The second step is an online contextual processing of users' requests. The request's terms are compared with the concepts' labels. Then, the user can navigate within the lattice to refine or generalize his/her query; he/she may also navigate within the semantic structure –thesaurus or ontology- if he/she needs domain knowledge.

This methodology was illustrated in the tourism domain. This research work was applied using a thesaurus but it will be extended with an ontology. The advantage of ontologies –or thesauri- for contexts is that they make it possible to link different contexts. The context may thus be widened by the ontology, as it provides precisions about a context. On the other hand, the advantage of contexts for ontologies is that relating ontologies to different contexts comes down to instantiating these ontologies in different contexts. We can see that as a way to populate ontologies.

In this paper, we showed how the use of conceptual lattices in conjunction with semantics could provide interesting results in the context of semantic and context-aware information retrieval. The combination of FCA and semantics can also be used for other purposes, e.g. to populate the ontology. It may also be applied to compare web sites through their respective lattices. The request's terms correspond to an entry point in one or more lattices. Then, the user may navigate from one lattice to others to refine his/her query. Another possible application is to help web site designers, as the lattice reflects the web site content: our method makes it easy to compare the resulting Web site/lattice with the original goals of the Web site designer.

References

1. McCarthy, J.: The advice taker. In: Minsky, M. (ed.) Semantic Information Processing, MIT Press, Cambridge (1968)
2. McCarthy, J.: Generality in Artificial Intelligence. Communications of ACM 30(12), 1030–1035 (1987)
3. Giunchiglia, F.: Contextual reasoning. Epistemologia, special issue on I Linguaggi e le Macchine, XVI:345–364 (1993)
4. Brézillon, P.: Context in Artificial Intelligence: I. A survey of the literature. Computer and Artificial Intelligence 18(18), 321–340 (1999)
5. Theodorakis, M., Spyratos, N.: Context in artificial intelligence and information modelling. In: Proceedings of the second Hellenic Conference on Artificial Intelligence (SETN 2002), Thessalonique (2002)
6. Guha, R.K., McCarthy, J.: Varieties of contexts. In: Blackburn, P., Ghidini, C., Turner, R.M., Giunchiglia, F. (eds.) CONTEXT 2003. LNCS, vol. 2680, pp. 164–177. Springer, Heidelberg (2003)
7. Dolog, P., Stuckenschmidt, H., Wache, H.: Robust Query Processing for Personalized Information Access on the Semantic Web. In: Larsen, H.L., Pasi, G., Ortiz-Arroyo, D., Andreasen, T., Christiansen, H. (eds.) FQAS 2006. LNCS, vol. 4027, pp. 343–355. Springer, Heidelberg (2006)
8. Mrissa, M., Ghedira, C., Benslimane, D., Maamar, Z.: A Context Model for Semantic Mediation in Web Services Composition. In: Embley, D.W., Olivé, A., Ram, S. (eds.) ER 2006. LNCS, vol. 4215, pp. 12–25. Springer, Heidelberg (2006)
9. Bouquet, P., Giunchiglia, F., Van Harmelen, F., Serafini, L., Stuckenschmidt, H.: Contextualizing Ontologies. Journal of Web Semantics 1(4), 1–19 (2004)
10. Doan, A., Madhavan, J., Domingos, P.: Learning to Map between Ontologies on the Semantic Web. In: The 11th International World Wide Web Conference (WWW 2002), Hawaii, May 7-11 (2002)
11. Priss, U.: Lattice-based Information Retrieval. Knowledge Organization 27(3), 132–142 (2000)
12. Messai, N., Devignes, M.-D., Napoli, A., Smaïl-Tabbone, M.: Querying a Bioinformatic Data Sources Registry with Concept Lattices. In: Dau, F., Mugnier, M.-L., Stumme, G. (eds.) ICCS 2005. LNCS, vol. 3596, pp. 323–336. Springer, Heidelberg (2005)
13. Safar, B., Kefi, H., Reynaud, C.: OntoRefiner, a user query refinement interface usable for Semantic Web Portals. In: Application of Semantic Web Technologies to Web Communities (ECAI 2004), 16th European Conference on Artificial Intelligence, Valencia, Spain, August 22-27, 2004. pp. 65–79 (2004)

14. Carpineto, C., Romano, G.: Exploiting the Potential of Concept Lattices for Information Retrieval with CREDO. Journal of Universal Computer Science 10(8), 985–1013 (2004)
15. Carpineto, C., Romano, G.: Galois: An order-theoretic approach to conceptual clustering. In: Proc. Of the 10th Conference on Machine Learning, Amherst, MA, pp. 33–40. Kaufmann, San Francisco (1993)
16. Wille, R.: Line diagrams of hierarchical concept systems. Int. Classif. 11, 77–86 (1984)
17. Birkhoff, G.: Lattice Theory, 1st edn., Amer. Math. Soc. Pub. 25, Providence, R. I (1940)
18. Barbut, M., Monjardet, B.: Ordre et classification, Algebre et combinatoire, Tome 2, Hachette (1970)
19. Godin, R., Chau, T.-T.: Incremental concept formation algorithms based on Galois Lattices. Computational intelligence 11(2), 246–267 (1998)
20. Wille, R.: Concept lattices and conceptual knowledge systems. Computers & Mathematics Applications 23(6-9), 493–515 (1992)

A Multi-representation Ontology for the Specification of Multi-context Requirements

Achraf Mtibaa and Faîez Gargouri

MIRACL Laboratory - Higher institute of data processing and multi-media of Sfax-Tunisia
achraf.mtibaa@issatgb.rnu.tn, faiez.gargouri@fsegs.rnu.tn

Abstract. There are many problems associated with requirements engineering. In fact, some concepts can be seen differently according to the user's context. The requirement specification generates several problems of incoherence, ambiguity and difficulty when users admit different contexts. These problems may lead to poor requirements and unsatisfactory or unacceptable future system. For this reason, we propose a multi-representation ontology to solve these conflicts and problems. In this paper, we propose our motivation and we expose our top-down approach for building a multi-representation ontology dedicated to multi-context requirements specification in the e-learning field.

Keywords: Ontology, Multi-representation, Requirement specification, Multi-context.

1 Introduction

User's requirements and applications' specificities have a strong influence on the system modelling. With the development of the Internet, the data are increasingly accessible and new requirements related on sharing and exchanging data appear. The complexity of the real world entities generates several possible interpretations or representations by various applications. It is thus necessary to define powerful data models to allow a coherent management for several contextual representations of the same real world entity. These models are called *multi representations models*, and consist of *preserving, within the same database, various representations of the same object* [1].

The standard IEEE 1220 defines Engineering System (ES) as *an interdisciplinary collaborative approach for the progressive development and the checking of a harmonious solution satisfying the user expectations and acceptable for the environment*. The purpose of Requirements Engineering (RE) is to lead to the realisation of the users' and customers' requirements for a future system. During the RE process, these requirements are elicited, negotiated, validated, specified and documented in requirements documents. Developing a quality system necessitates quality system requirements. Therefore, RE is an important phase of a system development project. Underestimating this importance may lead to aborted projects, overrun budgets, or system crashes, as numerous experiences have shown [2]. Whereas conventional methods were devoted to languages and models to express

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 259–269, 2009.

requirements, the RE community has perceived RE as a process that goes beyond the production of complete and consistent system requirements specifications. RE is also *a cooperative and iterative learning process* [3]. As an iterative learning process, RE relates closely to elicitation, validation, tracing, revision, specification and reuse. Moreover, as a cooperative process, requirements elicitation involves multiple stakeholders who have to communicate and understand each other: users, customers, domain experts, project managers, designers of previous system releases, etc. The new functionalities required by the system users appeared in the step of the Requirement Specification (RS). We view this latter (RS) as a declarative description of requirements that a system should provide. In RS, several problems can be generate such as lack of coherence, ambiguity and difficulty when users admit different contexts.

Ontologies are considered as one of the most significant domains in research area. The most referred definition of ontology is given in [4], an *ontology is an explicit specification of a conceptualization*. This conceptualization makes it possible to identify, using an abstraction process, the concepts and the main terms of a given domain.

In this paper, we propose a multi-representation ontology to solve conflicts and problems of the multi-context requirements specification. The rest of this paper is organized as follows. Section 2 gives our motivation to use the multi-representation ontology dedicated to the specification of multi-context requirements. Section 3 presents our top-down approach for building a multi-representation ontology dedicated to multi-context requirements specification and we start some steps of our approach in the e-learning field. Section 4 concludes the paper and alludes to our future works.

2 Motivation

The problem of multi-representation is encountered when a given concept can be seen in various prospects according to the context used. The multi-representation occupies in the ES a particular place for the following factors:

- Considerable increase of data volumes represented in various forms and generated from users when using varied contexts.
- Contextual sharing and data exchange between applications treating the different elements of the same system.
- Complexity of the data which reflects the extreme complexity of the real world.
- Multiple facets of the data which translate users' diversity.
- Dynamic requirement evolution.

Context can be defined as *the interrelated conditions in which something exists or occurs* [5]. The use of the word "context" tends to be unclear because everything in the world happens in a given context. Schmidt defines context as *knowledge about the user's and it device's state, including surroundings, situation, and to a less extent, location* [5]. However, for Dey in [6], it is *any information that can be used to characterize the situation of an entity*. Thus, combining several context values may generate a more powerful understanding of the current situation.

When a given domain allows only one context, it also allows one ontology and only one. Such an ontology is called, according to [7], a Mono-Representation Ontology (MoRo) or a mono-contextual ontology. In some cases, a given domain can have more than one ontology, where each one is related to a particular context. Thus, a concept is defined according to several contexts and to several representations; this ontology is called a contextual ontology. An ontology described according to several contexts at the same time is called, according to [7], a MUlti-Representation Ontology (MuRo). Thus, MuRo is defined, according to [8] and [7], as *an ontology characterizing the concepts by a variable set of properties (static and dynamic) or attributes in several contexts and several granularities*.

During our recent works [9], [10] and [11], we studied the ontology of the domain. This ontology defines a concept by a whole of properties, operations, structural and semantic relations with the other concepts of the concerned field. This definition of the concept is limited for only one context. For RS, a multitude of problems arise when the future users have many points of view of the same system. Each user has a particular context to specify his/her requirements. These various requirements are specified according to several contexts and so, they are multi- represented. Thus, we obtain several multi-context RS for the same system, modeled using various representations. Consequently, some multi-context and multi-representation conflicts can appear.

For RS, multi-representation leads to various problems:

- Data Duplication: the same requirement can be presented according to several contexts. Thus the classic problems raised by the data duplication emerge.
- Complexity to the modeling of the various representations of RS which are “Multi-User” and multi-represented.
- Difficulty to maintain the global requirements’ coherence: to maintain this coherence we need to integrate all the specified multi-context requirements.

As a solution, we propose the use of a multi-representation ontology, which will assist users thereafter to specify their multi-context requirements. Therefore, we justify the use of the MuRo by the following points [12]:

- Possibility to have more than one context when specifying the user’s requirements. Consequently, a concept has one or more definitions according to each context.
- Dynamic aspect of MuRo: we can add or modify, when requested, a given ontology concept according to another context.

In our work, multi-representation ontology will be used as a reference for users having different contexts in order to assist them when specifying their requirements according to different contexts. MuRo makes it possible to guide the future users to specify their system requirements and to solve conflicts resulting from the difference of contexts and points of view.

In the next section, we present our top-down approach for multi-representation ontology building dedicated to the specification of multi-context requirement.

3 Top-Down Approach for Multi-representation Ontology Building Dedicated to the Specification of Multi-context Requirement

Recently, several methodologies have been developed to engineer ontologies in a systematic and application driven way. However, when considering the needs of ontology engineers and ontology users various aspects of ontology engineering still need significant improvement (semi-automatic methods, design patterns, design rationales and provenance, economic aspects, etc.) [13]. In literature, the main works on multi-representation ontologies are in the geographic information system. In ES domain, from the problems detailed in the second section, we justify the multi-representation ontology use to resolve these problems of the multi-context requirements specification. In this section, we present our top-down approach for multi-representation ontology. The effort needed for engineering ontologies is up to now a major obstacle to developing ontology-based applications in commercial settings. Therefore, the tight coupling of manual methods with automatic methods is needed. Our approach is semi-automatic and is based on our work about the information system' ontology building [14] [15] [16].

Figure 1 presents our top-down approach. The starting point to build our ontology is a set of RS, for a given field, expressed using different formalisms and according to different contexts. As a first step, we identify the studied field and the different RS contexts. For that, in the second step, we propose a pre-treatment to identify the contexts and to determine the formalisms (textual, semi-formal or formal) used by the different RS. Then, we extract the requirements. We classify them into *actors*, *functionalities* or *relationships actor-functionality*. In the third step, we classify the extracted requirements by identifying the structural and semantic relationships between the concepts of the various contexts. This classification represents our first version of the Multi-representation ontology. This first version ($version_0$) is manually built. User help is possible during these three steps. From another RS (RS_i) according to another context ($context_i$), we update our Multi-representation ontology $version_{i-1}$ ($version_0$ for the first time). This enrichment will be made in the three following steps. The fourth step compares the concepts of Multi-representation ontology $version_{i-1}$ ($version_0$ for the first time) and the new concepts extracted from RS_i. Then, we deduce, from this comparison, the semantic relationships. Finally, we update our $version_{i-1}$ ontology by new concepts and their structural and semantic relationships, deduced by the last step. This updating generates the $version_i$ of our Multi-representation ontology. We can note that these three steps are semi-automatic and the users' interventions are possible to confirm each decision. They are also iterative and progressive. In the next section, we present our study field and we detail the three first steps of our proposed approach.

To clarify our work, we present the general architecture of our system shown in figure 2. We start from several (N) RS types for several (N) contexts. First, we will formalize each context. Then we realize modules for different formalisms (natural language, semi-formal or graphic documents and formal documents). We present, after, the multi-context requirement concepts with their structural and semantic relationships according to different contexts. These concepts and their relationships

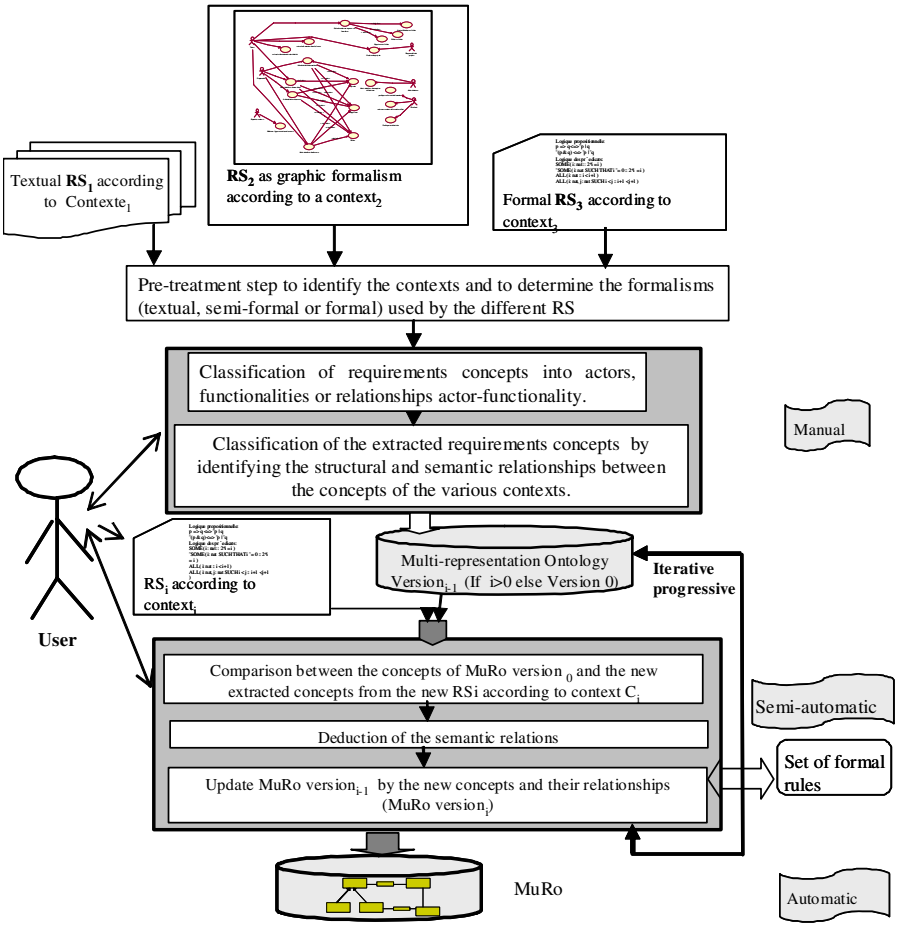


Fig. 1. Top-down approach for multi-representation ontology building

represent our MuRo. This latter, will be displayed to the future users and will assist the users to specify their future multi-context requirements. In the next section, we present our approach for building a MuRo.

In order to experiment our different proposals we take the E-learning field. In the following sub-sections, we first present the E-learning, its actors as well as their dynamic context diagram. We also point out, in the second sub-section, the use case diagrams and the Live Sequences Charts (LSC) and the corresponding examples for our studied field. In the third sub-section, we present briefly the manual phase of our top down approach.

3.1 E-learning Application

E-learning is defined, in [17], as interactive learning in which the learning content is available online and provides automatic feedback to the student's learning activities.

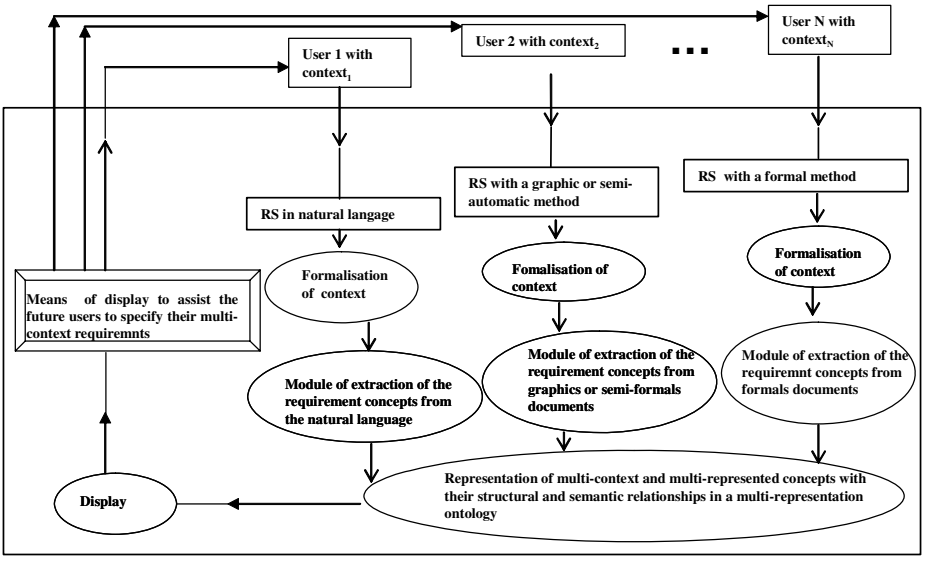


Fig. 2. General architecture of our system

Online communication with real people may or may not be included, but the focus of e-learning is usually more on the learning content than on communication between learners and tutors. In the glossary of *elearningeuropa.info*, e-Learning is defined as: the use of new multimedia technologies and the Internet to improve the quality of learning by facilitating access to resources and services as well as remote exchanges and collaboration.
The principal actors of the E-learning field are:

1 The Student: This actor must give its identification (password and login) to reach his profile and to consult the offered courses.
2 The Tutor: After giving his identification, this actor can create some courses and modules, follow the various types of discussion and answer to the students' questions. He can also publish his courses and put them on line.
3 The Administrator: This actor will have to manage all users' accounts and to create the various profiles.
4 The contents expert: This actor is responsible of what learners must have like formation as well as their evaluation.

The four actors, while connecting to the site, must be identified to consult their profiles and to pass in the corresponding space, according to their privileges. The identification step is a common operation for the four actors. The figure 3 gives the different actors' connections to the system expressed with a dynamic context diagram.

3.2 Use Case Diagrams and Live Sequence Charts

Before presenting the different multi-context and multi-representation RS, we present briefly, the use case diagrams and the live sequence charts (LSC) types.

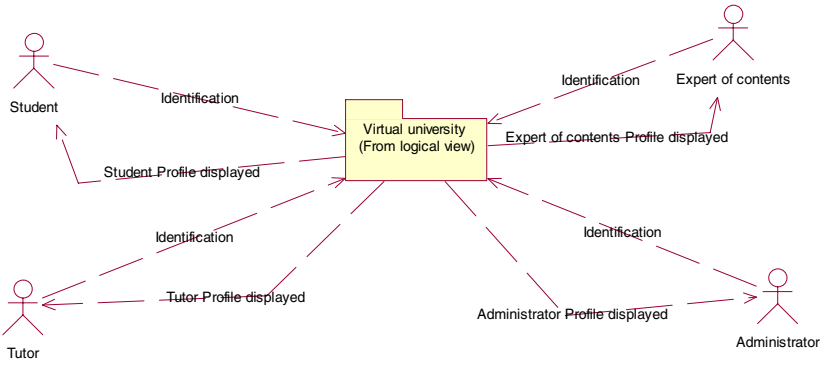


Fig. 3. Dynamic context diagram

Use case diagram consists of a set of use cases, each of which describing a service that the system will provide for some kinds of users called actors. It describes the functional requirement [18].

The LSC, have been proposed as an extension to Message Sequence Charts (MSCs) [19]. They consist of sets of processes, each denoted by rectangles containing process identifiers. Process lifelines extend downward from each process. Arrows represent messages passed from senders' process to receivers. The time scale of each process is independent of the others with the exception of the partial order imposed on the events occurring in the sending and receiving lifelines by a passed message. Sets of events that may occur in any order are marked by coregions, dotted lines near the lifeline on which the events occur.

Table 1. Textual formalism for the administrator RS context

TEXTUAL IDENTIFICATION	
Title:	Role of the administrator in our E-learning system
Goal:	To manage the system platform
Summary:	The administrator guaranteed the correct system functionality
Actors	Administrator, Tutor, Student, Expert of contents
SEQUENCES DESCRIPTION	
Pre-condition: The administrator must be authenticated for the system	
Sequence: * The administrator must be connected to the system by a login and a password [Exception1: loginfield] ***Sequence:*** * The administrator must: - To manage the students platform members (to manage the accounts, to manage the profiles, to manage the documents, to manage the inscriptions, etc.) - To manage the tutors platform members - To manage the forums of discussion - To manage the talk * The administrator makes the maintenance of the system * The administrator puts the results of the examinations on line	

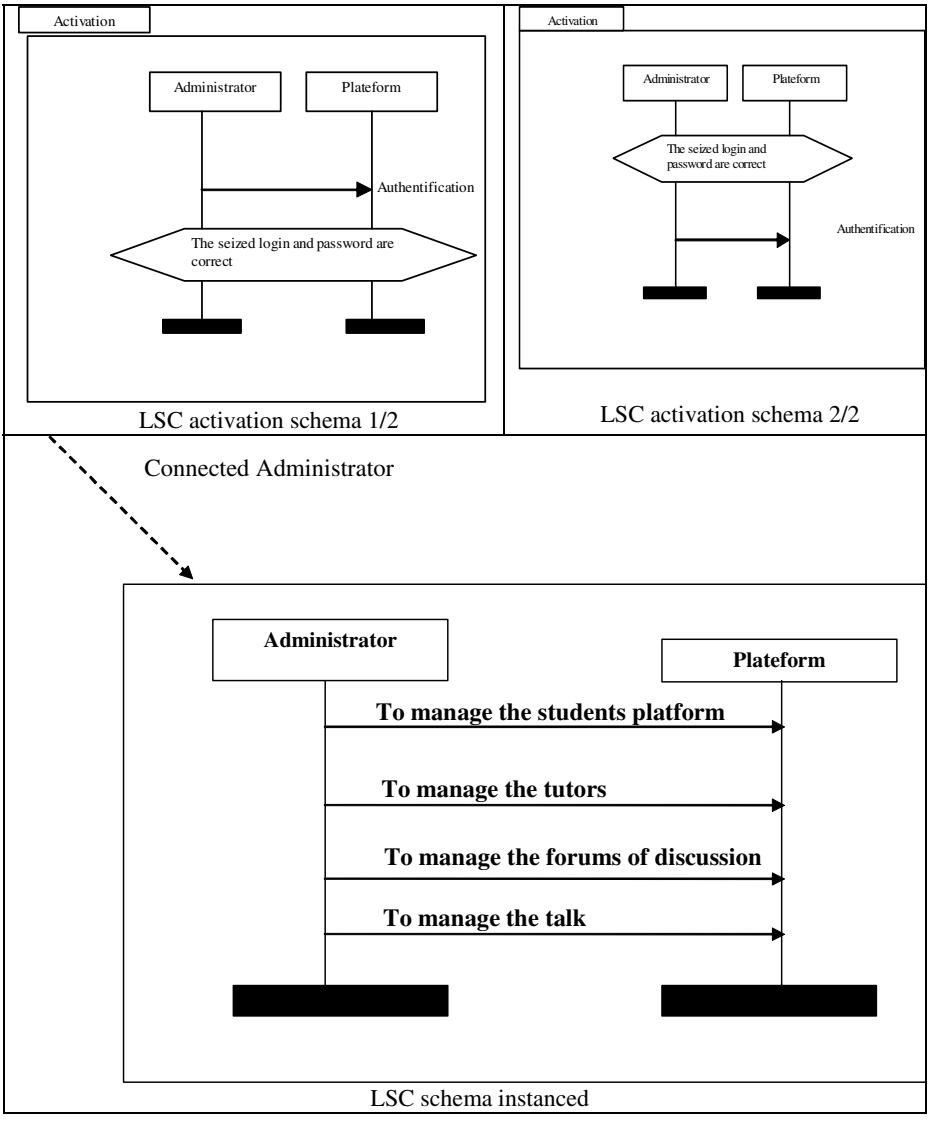


Fig. 4. LSC model of administrator context

For building our multi-representation ontology, its corpus should be collected. The corpus of our top-down approach is the set of concepts extracted from the different RS according to several actors' contexts. Once the corpus is ready, the first step for building Multi-representation ontology can be started.

For the E-learning field, table 1 presents the textual formalism for the *Administrator* RS context. Figure 4 gives a part of the LSC representation of the *Administrator* RS context, too. Figures 5 and 6 present two use case diagrams witch contain the different functionalities for the *Student* and *Tutor* contexts.

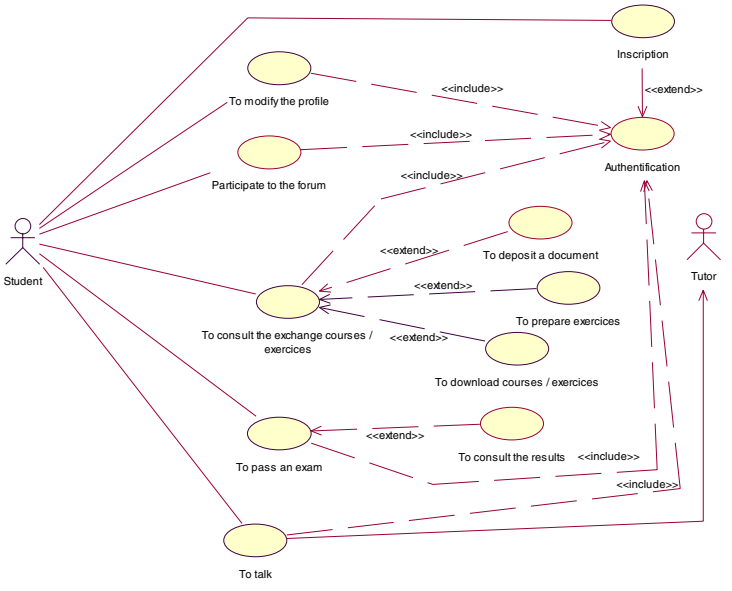


Fig. 5. Use cases diagram of student context

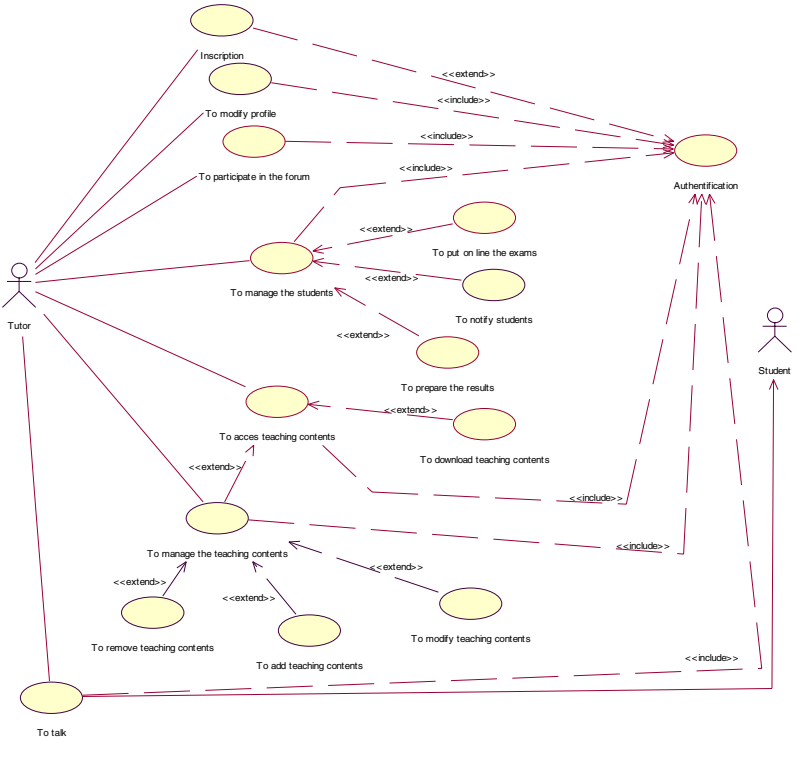


Fig. 6. Use cases diagram of tutor context

3.3 The Manual Phase of Our Top Down Approach

Each RS according to its formalism is different compared to the others. So, it is difficult to find similarities and differences between those RS' representations and many semantic conflicts can be appearing. For that, in our top-down approach we have proposed a pre-treatment step to identify the contexts and to determine the formalisms (textual, semi-formal or formal). We have three types of formalisms and four contexts for each actor. The second step of the manual phase of our top down approach is the extraction of the requirements concepts from each RS (presented in the last sub-section). After that, we classify these concepts into *actors*, *functionalities* or *relationships actor-functionality*, etc. The concepts extracted from different multi-context and multi-representation RS can be collected. After that, we add to these concepts the semantic relationships (*identity, synonymy, equivalence, antinomy, kind of*, etc) between these concepts. These concepts with their semantic and structural relationships represent our first version of our multi-representation ontology. This first version ($version_0$) is manually built. User help is possible during these three steps. We note that this manual phase is only for the first time.

4 Conclusion

In this paper, we proposed the use of a multi-representation ontology to solve the conflicts and problems of multi-context requirements specification. We outlined our top-down approach for building Multi-representation ontology. After that, we have detailed the manual phase of the proposed approach. This phase is experimented in the E-learning field.

We expect two research axes. The first axe aims to complete the semi-automatic and automatic phases for our top-down approach. The second one is to carry out an environment using our approach to ensure the adequacy between the multi-context user's requirements and the realization of a given system.

References

1. Bédard, Y., Bernier, E., Devillers, R.: La métastructure vuel et la gestion des représentations multiples, pp. 149–161. The Hermes Science Publications, Paris (2002)
2. Johnson, J.: Chaos: The Dollar Drain of IT project Failures. In: Application Development Trends, pp. 41–47 (January 1995)
3. Opdhal, A., Pohl, K.: Foundations of Software Quality. In: The Workshop Summary Proceedeings of the Fourth International Workshop on RE (REFSQ 1998), Pisa, Italy Presses Universitaires de Namur, pp. 1–11 (1998)
4. Gruber, T.: Towards principles for the design of ontologies used for knowledge sharing. The international Journal of Human and computer studies 43(5), 907–928 (1993)
5. Schmidt, A., Aidoo, K.A., Takaluoma, A., Tuomela, U., Van Laerhoven, K., Van de Velde, W.: Advanced interaction in context. In: Gellersen, H.-W. (ed.) HUC 1999. LNCS, vol. 1707, pp. 89–101. Springer, Heidelberg (1999)

6. Dey, A.K., Abowd, G.D.: Towards a Better Understanding of context and context-awareness. In: The technical Report GIT-GVU-99-22, Georgia Institute of Technology, College of Computing (June 1999)
7. Rifaieh, R., Arara, A., Benharkat, N.A.: MurO: A Multi-representation Ontology as a Foundation of Enterprise Information Systems. In: Das, G., Gulati, V.P. (eds.) CIT 2004. LNCS, vol. 3356, pp. 292–301. Springer, Heidelberg (2004)
8. Benslimane, D., Arara, A.: The multirepresentation ontologies: a contextual description logics approach. In: The 15th Conference on Advanced Information Systems Engineering, Klagenfurt/Velden, Austria, CEUR Workshop Proceedings 74 Technical University of Aachen (RWTH), June 16-20, 2003. Springer, Heidelberg (2003)
9. Mtibaa, A., M'Hiri, M., Gargouri, F.: Démarche de construction d'une ontologie pour la conception des systèmes d'information Cas du Commerce électronique. In: Cinquième Journée Scientifique des Jeunes Chercheurs en Génie Electrique et Informatique (GEI 2005), Sousse, Tunisia, Mach 25-27 (2005)
10. M'Hiri, M., Mtibaa, A., Gargouri, F.: OntoUML: Towards a language for the specification of information systems' ontologies. In: The Seventeenth International Conference on Software Engineering and Knowledge Engineering(SEKE) Taipei, Taiwan, Chine, July 14-16, pp. 743–746 (2005)
11. M'Hiri, M., Mtibaa, A., Gargouri, F.: Towards an approach for building information systems'ontologies. In: The Formal Ontologies Meet Industry (FOMI) Verona, Italy, June 9-10 (2005)
12. Mtibaa, A., Gargouri, F., Benslimane, D.: Spécification des besoins utilisateur basée sur une ontologie de multi-représentation. In: Sixième Journée Scientifique des Jeunes Chercheurs en Génie Electrique et Informatique (GEI), Hammamat – Tunisia, Mach 23-26, pp. 263–271 (2006)
13. Davies, J., Studer, R., Warren, P.: Semantic Web technologies trends and research in ontology-based systems. John Wiley & Sons, Europe, pp. 302–304 (April 2006) ISBN: 0-470-02596-4
14. Mhiri, M., Mtibaa, A., Gargouri, F.: OntoUML: Towards a language for the specification of information systems' ontologies. In: The Seventeenth International Conference on Software Engineering and Knowledge Engineering(SEKE) Taipei, Taiwan, Chine, July 14-16, pp. 743–746 (2005)
15. Mhiri, M., Mtibaa, A., Gargouri, F.: Towards an approach for building information systems'ontologies. In: The Formal Ontologies Meet Industry (FOMI) Verona, Italy, June 9-10 (2005)
16. Mhiri, M., Chabaane, S., Mtibaa, A.: Gargouri Faïez : An algorithm for building Information System's ontologies. In: The 8th International Conference on Enterprise Information Systems ICEIS 2006, Paphos, Cyprus, May 23-27 (2006)
17. Morten, F.-P.: E-learning.The State of the Art in the Work Package OneTHE DELPHI PROJECT NKI Distance Education (March 2003)
18. Xiaoshan, L., Zhiming, L., Jifeng, H.: Formal and Use-Case Driven Requirement Analysis in UML. Technical Report UNU/IIST Report No 230, UNU/IIST. International Institute for Software Technology, The United Nations University, P.O. Box 3058, Macau (March 2001)
19. Werner, D., Harel, D.: LSCs: Breathing Life into Message Sequence Charts. Formal Methods in System Design, pp. 45–80 (2001)

Scalability of Source Identification in Data Integration Systems

François Boisson, Michel Scholl, Imen Sebei, and Dan Vodislav

CNAM/CEDRIC, Paris, France
francois.boisson@gmail.com, {scholl,imen.sebei,vodislav}@cnam.fr

Abstract. Given a large number of data sources, each of them being indexed by attributes from a predefined set $\mathcal{A}$ and given a query q over a subset Q of $\mathcal{A}$ with size k attributes, we are interested in identifying the set of all possible combinations of sources such that the union of their attributes covers Q. Each combination c may lead to a rewriting of q as a join over the sources in c. Furthermore, to limit redundancy and combinatorial explosion, we want the combination of sources to produce a *minimal* cover of Q. Although motivated by query rewriting in OpenXView [3], an XML data integration system with a large number of XML sources, we believe that the solutions provided in this paper apply to other scalable data integration schemes. In this paper we focus on the cases where the number of sources is very large, while the size of queries is small. We propose a novel algorithm for the computation of the set of minimal covers of a query and experimentally evaluate its performance.

1 Introduction

The tremendous multiplication of data sources at every level (personal, enterprise, communities, web) in the past few years raises new challenges in querying such heterogeneous and highly distributed data.

In this context, we address a general problem when querying a large number N of data sources. Assume each source is indexed by attributes from a predefined set $\mathcal{A}$. Given a query q over a subset Q of $\mathcal{A}$ with k attributes, we want to find the sources that can partially or totally answer the query. If the source partially answers the query then it might, by a join with other sources, totally answer the query. Thus we are interested in identifying the set of all possible combinations of sources such that the union of their attributes covers Q.

The meaning of attributes is very general here. A data source indexed on an attribute provides data related to that attribute. In concrete cases attributes may be elements of a mediation schema connected to the data source through various mappings, or simple keywords that characterize the data source content.

Each combination of sources that covers Q may lead to a rewriting of q as a *join* over these sources. We consider here that joins between sources are realized in *a predefined way* and are not explicitly given in query q or in mappings. E.g., implicit joins between data sources may be fusion joins on key attributes [1,3], natural joins for relational sources [5], joins based on links, etc.

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 270–279, 2009.

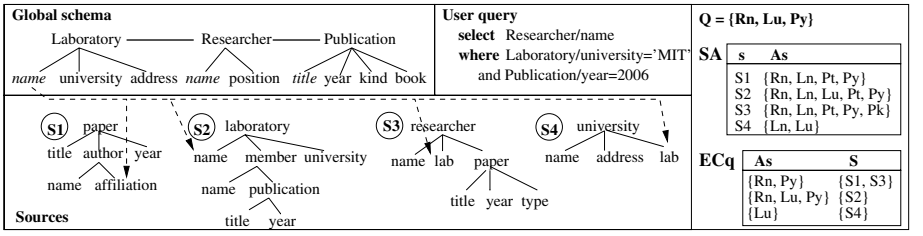


Fig. 1. An example of data integration system

Furthermore, the number of possible combinations of sources being potentially very large, we made the choice to limit combinatorial explosion and redundancy by only considering combinations that produce *a minimal cover* of Q. In a minimal cover, no source can be removed without breaking the cover of Q, i.e. each source covers some attribute in Q that no other source in the combination covers. Intuitively, the more attributes of a query are found in a source, the more pertinent the source is for that query. Minimal covers limit dispersion of query attributes among sources, which would produce less pertinent results.

This work on source identification was motivated by data integration systems dealing with a large number of (relational or XML) local-as-view [8] sources, where queries written on a common schema have to be translated into queries on the sources. Searching by keywords on the web is another class of applications that we have in mind. The idea is to extend current search engines, that limit answers to single documents containing (part of) the keywords so as to consider joins based on hyperlinks between documents.

More specifically, this work was initiated when studying the query rewriting algorithm of OpenXView [3], a data integration model for a large number of XML sources. In OpenXView a query expressed on an ontology is translated into a union of joins of pattern tree queries on the sources (each source may have a different DTD or XML Schema), thanks to a mapping between ontology concepts and XML elements.

Although the mechanism studied in this paper applies to a large variety of mediation schemes, we chose to illustrate it on an example of OpenXView data integration, shown in Figure 1. Sources S_1-S_4 contain XML documents about research laboratories, researchers and their publications, whose schemas are presented in a tree-like form. The global schema is an ontology with concepts *Laboratory*, *Researcher* and *Publication*, each one with several properties that form a tree, those in italic being *key* properties used for implicit joins (e.g. *name* for *Researcher*). In this case, the set of attributes $\mathcal{A}$ is the set of all properties (paths) in the global schema: $\mathcal{A}$={*Laboratory/name, Laboratory/address,* ...}. The indexing of sources with attributes is realized through node-to-node mappings between attributes and source schema elements, e.g. *Laboratory/name* is mapped to *paper/author/affiliation* in S_1, to *laboratory/name* in S_2, etc. For the sake of readability, only a few mappings are shown.

An OpenXView query q is a set of projections and selection conditions over elements of $\mathcal{A}$. The query in Figure 1 asks for names of researchers from MIT that

published in 2006; the set of query attributes is $Q = \{$*Researcher/name, Laboratory/university, Publication/year*$\}$. The source identification problem is to find all combinations of sources in S_1-S_4 that produce a minimal cover of Q. E.g., combination (S_1, S_4) produces a minimal cover of Q, because S_1 covers the researcher name and the publication year, while S_4 covers the remaining attribute in Q (university name), but (S_1, S_2) is not minimal, because S_2 covers all attributes in Q and S_1 becomes redundant. Note that source identification is just the first step in query rewriting prior to actual query translation. This paper focuses on this critical step. We address cases where the number of sources may be very large (1000 or more), while the query size is small (no more than 8 attributes). In a previous paper [3], we designed a simple algorithm, referred to in the following as SI_1, for identifying the relevant sources, which scans all the sources and progressively constructs the minimal combinations. In the following, we propose an efficient and scalable algorithm for source identification, called SI_2, based on the pre-computation of minimal covers and appropriate when the number of query attributes k is ≤ 8, which in most applications is a reasonable assumption. We experimentally compare the time performance of this algorithm and that of algorithm SI_1 and show that SI_2 drastically outperforms SI_1. However SI_2 requires a significant amount of space for storing the pre-computed sets of minimal covers.

The paper is organized as follows. Section 2 discusses related work, then Section 3 presents both algorithms SI_1 [3] and SI_2. Section 4 reports on our performance study and some conclusions are drawn in Section 5.

2 Related Work

We focus on the integration of a *large* number of sources, where publishing of new sources is frequent and unpredictible. We deal with cases where implicit joins between sources exist, hence cases with few restrictions on combining sources to answer the query. Moreover, we aim at simplifying the expression of queries by the final user: querying with selection and projection of attributes, which does not introduce strong constraints for combining sources. We limit the number of combinations to a subset with reasonable size: *minimal* combinations. Our goal is to find fast algorithms for computing them, scalable with the number of sources N. Note that we could rank query rewritings obtained from minimal combinations. This is out of the scope of this paper. Although source identification is a well known issue in querying data integration systems, to our knowledge, existing algorithms for source selection are not appropriate in our context:

In global-as-view systems [6,2,7], joins between sources are defined in mappings, which makes source selection trivial, but maintaining mappings when adding new sources is difficult. Attempts to improve maintainability, such as that proposed in [11], where joins are moved to an intermediate level of the global schema, are not enough to make global-as-view systems appropriate for large scale integration with frequent changes.

Local-as-view architectures better match our needs. But existing systems are based on specific models and come with limitations when adapted to our

context. The best known query rewriting algorithms are Bucket [5] and its optimization MiniCon [10], designed for relational integration systems. In these algorithms, not scalable with N, even if implicit joins between sources exist, the query model introduces strong constraints for source combinations. If the general Bucket strategy can be adapted to our context ([3] shows that our SI_1 algorithm outperforms a scalable variant of Bucket), MiniCon is too dependent of its underlying model, to be reused for other data integration models.

Among the other local-as-view approaches, [12] proposes an integration framework for XML and relational data which does not fit our context since joins are not implicit, but expressed in the global schema. [4] studies the contribution of sources to a given query in integration systems, i.e. their classification as self-contained or complementary to other sources, but does not compute source combinations. In contrast, [1] uses implicit joins based on keys, but its recursive algorithm does not scale with N, nor considers minimality.

3 The Algorithms

3.1 Introduction and Notations

Let $\mathcal{S}$ denote the set of sources whose size is N. Indexing of data sources with attributes of $\mathcal{A}$ is represented by (N1NF) relation SA. SA is a set of couples $[s, A_s]$, where $s \in \mathcal{S}$ is a source and A_s is the subset of attributes in $\mathcal{A}$ provided by s. For the example in Figure 1, the subset of attributes of the query Q and SA are shown on the right side of the figure. We use an abbreviated notation for the attribute names, e.g. *Rn* (*Py*) stands for *Researcher/name* (*Publication/year*).

Both source identification algorithms presented in this section have two steps:

1. The first step, common to both algorithms, takes as inputs the query q and $SA(s, A_s)$ and gives as an output $EC_q(A_s, S)$ the set of couples where the first component $A_s \subseteq Q$ is a non-empty subset of attributes of the query and the second component $S \subseteq \mathcal{S}$ is the subset of sources providing *exactly* the attributes in A_s.

   There are at most $2^k - 1$ tuples in relation EC_q where k is the number of attributes in the query. Basically for a query, the set of sources is split into equivalence classes according to the attributes provided by the source. Tuples of EC_q represent all non empty equivalence classes for query q.

   The $O(kN)$ algorithm scans the N sources in SA and for each $[s, A_s] \in SA$, it computes $A_q = A_s \cap Q$ (in $O(k)$ using a hash table for A_s). If $A_q \neq \emptyset$, source s is added to EC_q in the set of sources corresponding to the set of attributes A_q (constant time if $EC_q(A_s, S)$ is represented as a hash table with key A_s).

   For the example in Figure 1, the intersection with Q of attribute sets in SA produces three distinct sets, each one accounting for a tuple in EC_q. E.g., source S_4 covers only attribute *Lu* in Q, while S_1 and S_3 cover both exactly *Rn* and *Py*, thus belong to the same equivalence class. Compared to the number of sources N which might be larger than one thousand, the

```
SI1(EC_q, Q)
 output : set of minimal covers
Begin
  c_0 = ( )
  E_0 = π_{A_s}(EC_q)
  MA_0 = ∅
  return MC(c_0, E_0, MA_0, Q)
End SI_1
```

```
MC(c, E, MA, Q)
 output : set of minimal covers
Begin
 If (MA ≟^s Q) return({c})
  C=∅
 While E ≠ ∅ repeat
   i=pop(E)
   If minimal(c, i, MA)
     C=C ∪ MC(c+i, copy(E),
          MA ∪ Attr(i), Q)
   End If
 End While
 return C
End MC
```

```
minimal(c, i, MA)
 output : boolean
Begin
 If Attr(i) ⊂ MA
   return(false)
 End If
 NewMA=MA ∪ Attr(i)
 For each m ∈ c repeat
   If (NewMA − Attr(m))≟^s NewMA
     return(false)
   End If
 End For
 return true
End minimal
```

Fig. 2. SI_1 algorithm with recursive minimal covering (MC) and minimality test

maximal number of equivalence classes keeps small for reasonable values of the number of attributes in a query k (with $k = 6$ it is at most 63).

2. The second step takes as an input EC_q and computes $\mathcal{C}$ the set of minimal covers of sources (mcs). An mcs is a cover of the attributes in Q, i.e. a set of one or more classes (tuples of EC_q) such that the union of the query attributes in each element (*member*) of the set is equal to Q. Furthermore as aforementioned, we require this cover to be minimal, i.e. if we remove any member of the mcs, then Q is not anymore covered by the attributes in the mcs. Once the set of mcs has been computed then the source identification is over: each mcs represents a possible combination of sources. Let $m_1, ..., m_n$ be the n members of an mcs. Then $s_1, ..., s_n$ where $s_i \in m_i$ for $i \in \{1, ..., n\}$, is a potential combination of sources for query translation.

   Since for a given query with size k attributes, $|\mathcal{C}_k|$, the maximal number of mcs, is exponential in k[1], one expects this second step to be expensive.

We present below two algorithms for step 2 and compare their performance in the following section.

3.2 Algorithm SI_1

This algorithm (Figure 2), introduced in [3], recursively generates all minimal covers of Q with equivalence classes in EC_q. The recursive algorithm MC takes as inputs, a minimal partial cover c (to be completed to a minimal cover), the ordered set of equivalence classes E not yet considered in building c, the multiset MA of query attributes covered by c (counting for each attribute the number of times it is covered), and the target query attributes Q to cover.

The initial call to MC starts with an empty cover and with all the equivalence classes extracted from EC_q. Each call to MC tests whether c is a new solution

[1] The number of minimal set covers of the set of integers K={1..k} is [9]:
$| \mathcal{C}_k | = \sum_{\ell=1}^{\ell=k} \mu(k, \ell)$, where $\mu(k, \ell)$ is the number of minimal covers of K={1,..,k} with ℓ members:
$\mu(k,\ell) = \frac{1}{\ell!} \sum_{m=\ell}^{min(k,2^\ell-1)} C_{m-\ell}^{2^\ell-\ell-1} m! s(k,m)$,
where s(k,m) is a Stirling number of the second kind. E.g.: $| \mathcal{C}_4 | = 49$, $| \mathcal{C}_6 | = 6424$, $| \mathcal{C}_8 | = 3732406$.

```
SI2(members, MSCk)
     output: valid, all bits initially set to 1
Begin
    For each As in P({1, ..., k}) − {∅} loop
      If members(As)=false
         valid = valid and ¬ MSCk(As)
      End If
    End For
End SI2
```

Fig. 3. Algorithm SI_2 utilizing pre-computed minimal set covers

by comparing Q and MA ("$\stackrel{s}{=}$" denotes set equality). If not, one tries to extend c with each element of E, and if the new cover is minimal (tested with function *minimal*), it recursively calls MC. The minimality test verifies first if class i, to be added to partial cover c, is redundant (i.e., already covered by c). If not, it tests for each member of c if it becomes redundant when adding i to c. Tests use MA and $NewMA$, the multisets of attributes covered by c, respectively $c + i$.

In the example in Figure 1, algorithm SI_1 computes minimal covers of Q with equivalence classes of EC_q. The partial cover containing class $\{Rn, Py\}$ cannot be extended with class $\{Rn, Lu, Py\}$ because the new cover is not minimal ($\{Rn, Py\}$ becomes redundant), but can be extended with class $\{Lu\}$, which produces a minimal cover solution. The minimal covers in our example are c_{s1}=($\{Rn, Py\}$,$\{Lu\}$) and c_{s2}=($\{Rn, Lu, Py\}$). Given the sources in each equivalence class, c_{s1} produces combinations of sources (S_1, S_4) and (S_3, S_4), while c_{s2} produces only (S_2).

3.3 Algorithm SI_2

In contrast to algorithm SI_1, the novel algorithm we present in this section supposes the existence of the following structure: $MSC_k(A_s, c)$ a pre-computed relational table describing all the minimal covers of the set {1,2,...,k}. Tuple $[A_s, c]$ in MSC_k states that $A_s \in \mathcal{P}(\{1, 2, ..., k\}) - \{\emptyset\}$ is a member of minimal cover c[2]. As aforementioned[1], the number of minimal covers $|\mathcal{C}_k|$ of $\{1, ..., k\}$ increases very rapidly with k. We assume in the following that this table or its compacted N1NF version $MSC_k(A_s, C)$ where C is the list of minimal covers that include A_s as a member, holds in memory. This implies, with our implementation, an upper bound on k which is ≤ 8. Because of space limitations we do not detail the off line construction of $MSC_k(A_s, C)$: it is pre-computed, independently of any query, by an algorithm similar to algorithm SI_1.

Given MSC_k and EC_q (computed in step 1), the following relational expression calculates $\mathcal{C}_q$ the set of minimal covers of Q, which is composed only of members appearing in EC_q (equivalence classes):

$R(A_s) = \pi_{A_s}(MSC_k(A_s, c)) - \pi_{A_s}(EC_q(A_s, S))$

$S(c) = \pi_c(MSC_k(A_s, c) \bowtie R(A_s))$

$\mathcal{C}_q(c) = \pi_c(MSC_k(A_s, c)) - S(c)$

[2] Without loss of generality, attributes names in Q are replaced by an integer in $\{1, ..., k\}$.

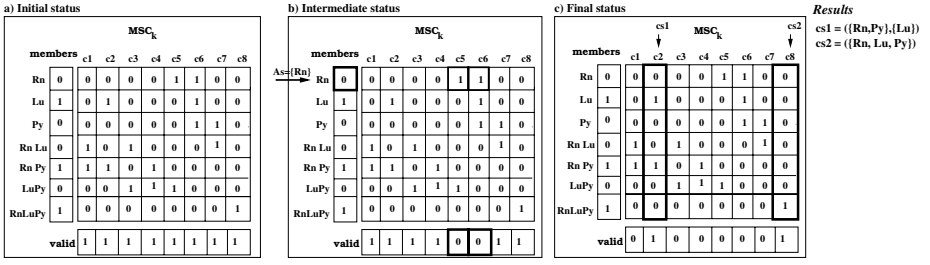


Fig. 4. SI_2 data structure

$R(A_s)$ is the set of "bad" members, where a bad member is a member of a minimal cover that is not in EC_q. Then $S(c)$ is the set of bad minimal set covers i.e. covers which hold at least one member in $R(A_s)$. Therefore $\mathcal{C}_q(c)$ is the set of "good" minimal set of covers i.e. made of members appearing in EC_q.

The algorithm in Figure 3 implements the computation of $\mathcal{C}_q(c)$. It relies on two boolean vectors:

1. *members* with size 2^k-1 where *members(*A_s*)* is set to 0 if the set of attributes A_s is in $R(A_s)$: it is a bad member. Note that vector *members* is constructed with no additional cost when building $EC_q(A_s, S)$ from the initial table $SA(s, A_s)$ which, recall, associates with each source its attributes.
2. *valid* with size the number of minimal set covers $|\mathcal{C}_k|$ where *valid(c)* is set to 1 if minimal set cover c is a good one: it belongs to $\mathcal{C}_q(c)$. All bits of *valid(c)* are initialized to 1.

MSC_k is implemented as a bitmap with $2^k - 1$ lines and $|\mathcal{C}_k|$ columns: $MSC_k(A_s, c)$ is set to 1 if cover c has for a member A_s.

Figure 4 displays for the example of Figure 1 the value of MSC_k and that of *members* and *valid* prior to the execution (Figure 4.a) after the first loop iteration (Figure 4.b) and at the end (Figure 4.c). With $k = 3$, there are 8 minimal covers and the final minimal covers are: $c_2 = (\{Rn, Py\}, \{Lu\})$, $c_8 = (\{Rn, Lu, Py\})$.

The size of the bitmap $|\mathcal{C}_k|$ drastically increases with k. Therefore k must be reasonably small not only because MSC_k must hold in memory but also because of the complexity of the bitmap operations is $O(|\mathcal{C}_k|)$, which grows very fast. For the amount of RAM memory available in today computers, the bound is $k \leq 8$ which is reasonable for the applications we have in mind. The complexity of algorithm SI_2 is $O(2^k) \times |\mathcal{C}_k|$. However since k is small the bitmap operation is fast. This is confirmed by our experiments reported in the following section.

4 Experiments

4.1 Experimental Environment

The algorithms and experiments were implemented in Java (JDK1.5.0-06), on a PC with 512 MBytes of RAM and a P4 3 GHz CPU, running Windows XP. We

use a synthetic set $\mathcal{A}$ of 300 attributes and a number of sources ranging from $N = 100$ to $N = 10000$. Assignment of attributes to sources (source indexing) follows a scheme motivated by the OpenXView model, but common to most data integration system, in which attributes are grouped in common structures (concepts in OpenXView, relations in relational models, etc). In our case, the 300 attributes correspond to 30 concepts of 10 attributes (properties) each. Source indexing and querying respect *locality*, i.e. we select attributes in several concepts instead of randomly choosing them in $\mathcal{A}$. More precisely, we used the following repartition for source indexing: 25% of the sources cover 2 concepts, 50% 3 concepts and 25% 4 concepts, with 50% of the concept's attributes (randomly chosen) in each case. Queries were randomly generated with k= 2, 4, 6 attributes randomly chosen within two concepts. The results presented in the following section are robust wrt changes in the repartitions and parameters. Because of space limitations we present only the results for the above synthetic set and queries.

4.2 Experimental Results

All algorithms were run in central memory. Our performance evaluation focuses on the total time for running each of the two steps of the algorithms for source identification. Each time measure is the average over 50 random queries with the same value of k and N. We successively made three experiments: the first one evaluates step 1 which is common to both algorithms. The second one compares the performance of both algorithms for step 2 and the third experiment studies the predominance of one step over the other depending on the algorithm. Because of space limitations we only present the most illustrative figures for the general behavior of our algorithms.

Step 1. Figure 5.a displays the number of equivalence classes (number of tuples of EC_q) vs the number of sources, for queries of size $k = 6$. Some nodes are labelled by the time taken to perform step 1. The experimental results confirm that the number of classes (which theoretically can reach 2^k-1), is much smaller in practice (for $N = 4000$, a very large number of sources, the average number of classes is 33). Anyhow this number is very small wrt N for all values of N. Recall that step 2 takes as an input classes and not the sources themselves. Step 1 reduces the source identification problem to the scanning in step 2 of an expected small number of classes. In contrast, naive algorithms such as Bucket [5] compute combinations directly from the sources. As expected, the time growth is linear in N. Experiments for other values of k confirmed the same behavior and the linearity with k.

Step 2. We successively study the impact of k and of N on the time performance of step 2. Figure 5.b displays the time for performing step 2 with both algorithms vs the number of attributes k, for 4000 sources. While with SI_2 it is almost negligible for all values of k [3], with SI_1 it increases drastically. Figure 5.c and Figure 5.d display the time for running step 2 respectively versus the number

[3] Recall that for $k > 8$ the amount of memory space required by this algorithm is prohibitive.

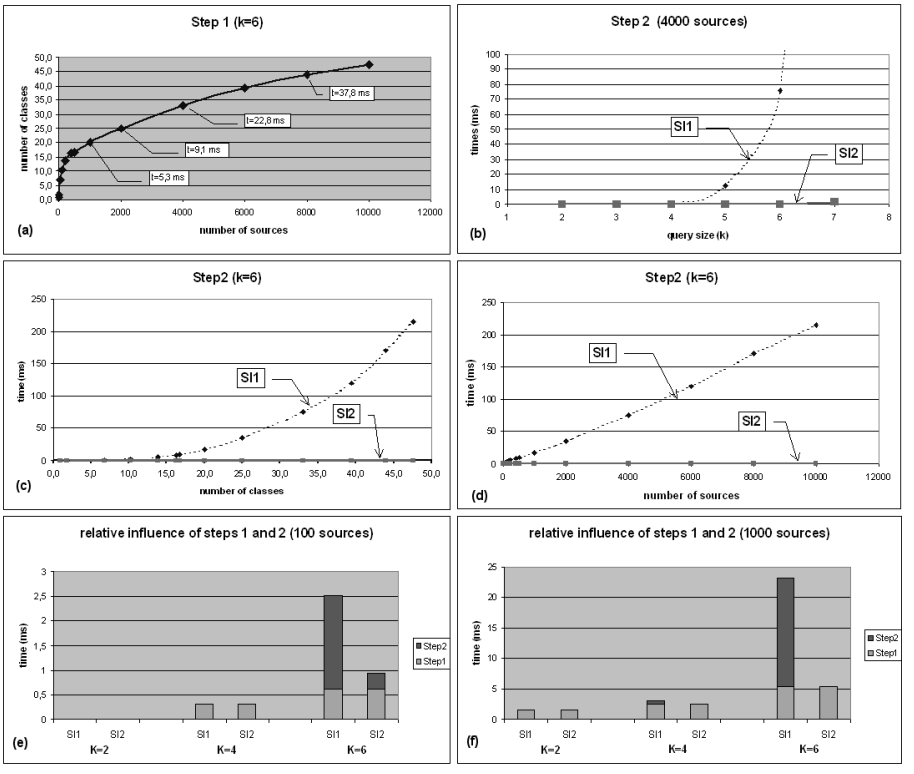


Fig. 5. Experiments

of sources (Figure 5.c) and the number of classes (Figure 5.d). They confirm that algorithm SI_2 drastically outperforms algorithm SI_1. Time for running step 2 with algorithm SI_2 is negligible or at least less than 1 ms even for a large number of sources (classes). In contrast, algorithm SI_1 performance degrades exponentially with the number of classes (Figure 5.c). Note however the quasi-linearity with N, because of the slow growth of the number of classes with N (Figure 5.d).

Step 1 / Step 2. Last, we study the relative importance of step 1 wrt that of step 2. Time versus k is plotted for 100 sources (Figure 5.e) and for 1000 sources (Figure 5.f). The first important observation is that with algorithm SI_2, when the number of sources is large, the time for running step 2 is negligible and therefore the source identification performance is that of step 1. As a matter of fact it was observed that as expected, the latter time is linear in k and N. Although we were not able to find sub-linear algorithms, step 1 still deserves some improvements. The second observation is that with algorithm SI_1, the larger k, the larger the predominance of step 2 duration over step 1, even for a relatively small number of sources: while with $k = 4$ step 2 is almost negligible, with $k = 6$, step 2 already lasts four times longer than step 1. Since step 2 duration is exponential with k, it is to be expected that with larger values of

k, the performance of source identification with algorithm SI_1 is that of step 2 which is exponential with k. Therefore, for this algorithm as well, values of k larger than 8 are unpractical.

In conclusion, we assessed the impact on performance of the organization of sources into equivalence classes depending on the query (step 1) and demonstrated the significant improvement on performance brought by algorithm SI_2 wrt to algorithm SI_1 proposed in [3]. SI_2 performs well even with a large number of sources.

5 Conclusion

This paper was devoted to source identification when a query with k attributes (keywords, properties, etc.) has to be translated into queries over a number of sources. This problem is central to query rewriting in data integration systems. We exhibited a new efficient algorithm and evaluated its performance. We shown that this algorithm is scalable and applies to environments with a very large number of sources. However this algorithm -as well as a comparable algorithm [3]- is not scalable with the number of attributes in a query. We evaluated this well known exponential behavior and gave for current technology, an experimental bound of 8 for the number of attributes in a query.

References

1. Amann, B., Beeri, C., Fundulaki, I., Scholl, M.: Querying xml sources using an ontology-based mediator. In: Meersman, R., Tari, Z., et al. (eds.) CoopIS 2002, DOA 2002, and ODBASE 2002. LNCS, vol. 2519, pp. 429–448. Springer, Heidelberg (2002)
2. Baru, C.K., Gupta, A., Ludäscher, B., Marciano, R., Papakonstantinou, Y., Velikhov, P., Chu, V.: XML-Based Information Mediation with MIX. In: SIGMOD (1999)
3. Boisson, F., Scholl, M., Ssebei, I., Vodislav, D.: Query rewriting for open xml data integration systems. In: IADIS WWW/Internet (2006)
4. Deutsch, A., Katsis, Y., Papakonstantinou, Y.: Determining source contribution in integration systems. In: PODS (2005)
5. Halvey, A.: Answering queries using views: A survey. The VLDB Journal, 270–294 (2001)
6. Josifovski, V., Schwarz, P., Haas, L., Lin, E.: Garlic: a new flavor of federated query processing for DB2. In: SIGMOD (2002)
7. Lenzerini, M.: Data integration: a theoretical perspective. In: PODS (2002)
8. Levy, A., Mendelzon, A., Sagiv, Y., Srivastava, D.: Answering queries using views. In: PODS (1995)
9. Macula, A.J.: Covers of a finite set. Math. Mag. 67, 141–144 (1994)
10. Pottinger, R., Halevey, A.: Minicon: A scalable algorithm for answering queries using views. The VLDB Journal, 182–198 (2001)
11. Vodislav, D., Cluet, S., Corona, G., Sebei, I.: Views for simplifying access to heterogeneous XML data. In: CoopIS (2006)
12. Yu, C., Popa, L.: Constraint-based xml query rewriting for data integration. In: SIGMOD (2004)

Semantic Annotation of Web Pages Using Web Patterns

Milos Kudelka, Vaclav Snasel, Ondrej Lehecka,
Eyas El-Qawasmeh, and Jaroslav Pokorný

Computer Science Dept., VSB – Technical University of Ostrava, Czech Republic
Computer Science Dept., Jordan University of Science and Technology, Irbid, Jordan
Department of Software Engineering, Charles University of Prague, Czech Republic
kudelka@inflex.cz, {ondrej.lehecka,vaclav.snasel}@vsb.cz,
eyas@just.edu.jo, pokorny@ksi.ms.mff.cuni.cz

Abstract. This paper introduces a novel method for semantic annotation of web pages. We perform semantic annotation with regard to unwritten and empirically proven agreement between users and web designers using web patterns. This method is based on extraction of patterns, which are characteristic for a particular domain. A pattern provides formalization of the agreement and allows assigning semantics to parts of web pages. We will introduce experiments with this method and show its benefits for querying the web.

1 Introduction

Semantic annotation of web pages concerns adding formal semantics (metadata, knowledge) to the web content for the purpose of more efficient access and management. Currently, the researchers are working on the development of fully automatic methods for semantic annotation (see, e.g., [2]).

For our research we consider semantic annotation and tracing user behaviour in the query-answering dialog: (1) to simplify querying; and (2) to improve relevance of answers. In the field of Internet search, we introduce a new perspective, which connects both goals in a native way. The key aspect of our perspective is smart focusing on the user and his expectations when searching information on the web. To be able to do this we need the user to share his expectation with us.

A simpler way is to turn our questions to professional web sites designers. Their primary mission is to fulfil user's expectation. A proof of this is that high-quality web pages and web solutions are widely accepted by users. Professional web designers apply practices, which come up from user's experiences. These practices relate with human sensation and allow simple orientation in supplied information. Solutions of the same problem are solved by different developers differently but at a certain level solutions are all the same. Similar web pages contain similar components. We can define this conformity such that there are similar web page components on the pages within the same application domain. These components are designated as web patterns.

Our method for semantic annotation of web pages is performed with regard to unwritten and empirically proven agreement between users and web designers. This method is based on extraction of patterns, which are characteristic for a particular domain. A pattern provides formalization of the agreement and allows assigning

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 280–291, 2009.

semantics to parts of web pages. We will introduce experiments with this method and show its benefits for querying the web.

Section 2 contains a short description of related works. Section 3 presents the patterns basics. Sections 4 and 5 present goals of our research. In Sections 6 and 7 we describe preparation of experiments and an analysis of results. Finally, Sections 8 and 9 focus on our future work and conclusions.

2 Related Work

There are two trends in the field of semantic analysis of the web today. One of them provides mechanism to semiautomatic (or manual) page annotation using ontology description languages and creation of semantic web documents [23], [10]. The second approach prefers an automatic annotation. In [3] there is a methodology based on a combination of information extraction, information integration, and machine learning techniques. The complex example of an automatic approach is the ambitious KIM project [15]. An application that performs automated semantic tagging of large corpora is described in [5]. It is based on the Seeker platform for large-scale text analysis. It marks up large numbers of pages with terms from a standard ontology.

Across all directions it is possible to see the use of ontology-based mechanisms, in the case of second approach along with knowledge bases. Our view on the problem of semantic analysis is in many cases similar to the presented techniques and tends to the automatic approach of semantic annotation. However our motivation of what and how to annotate is different. It seems to be on a higher abstraction level because we do not work with the semantics in meaning of the content of document, but more likely with the form of the document, which is related to the content and chosen domain.

The approach, which is similar to ours, is mentioned in [16]. It uses TXL language to analyze chosen parts of pages to obtain structured domain specific information (tourism domain). Other similar approaches are automatic transformation of arbitrary table-like structures into knowledge models [21], formalized methods of processing the format and content of tables to relevant reusable conceptual ontology [25] or domain-oriented approach to web data extraction based on a tree structure analysis [22]. Paper [1] presents techniques for supervised wrapper generation and automated web information extraction. The system generates wrappers, which translate relevant pieces of HTML pages into XML. The next similar approach to web data extraction is described in [18]. The principal component is a document schema (found in HTML code). Document schemata are patterns of structures embedded in documents.

On the side of query expression analysis and construction there are approaches, which are helpful for the user when formulating his requirement. One of the possible approaches is to use a cognitive search model [27]. There is a web search system prototype based on ontology that uses a cognitive model of the process of human information acquisition. Another way is to help user with specification of query based on interaction with user using genetic algorithms and fuzzy logic, e.g., [4], [17], [11].

There is interesting conjunction with paper [12] whose authors analyze web pages focusing on web site patterns. In three time intervals authors observed how web designers have changed web design practices. They also realized that content of web pages remains the same whereas form is being developed so it better fulfils user's expectation. Our work confirms results mentioned in the paper. Important for us are such web pattern characteristics that are independent of a web page design.

3 Patterns Basics

According to [24] patterns are structural and behavioural features that improve the applicability of software architecture, a user interface, a web site or something another in some domain. They make things more usable and easier to understand.

GUI patterns supply a solution of typical problems within design of user interface. Typical examples are organization of user controls into lists or tabs and so on. GUI patterns describe on a general level how to make structure of information within user interface. They also tell which components to use, how they should work together and how to work with them (see [24], [9] for examples).

In [24] there is a set of idioms which give us a general answer to question which types of user interfaces we can come across (Forms, Text editors, Graphic editors, Spreadsheets, Browsers, Calendars, Media players, Information graphics, Immersive games, web pages, Social spaces, E-commerce sites). For our purposes we focused on Calendars, Web pages, Social spaces, and mainly on E-commerce sites.

In this paper we have chosen selling products domain for our experiments. Patterns identified for other domains are, e.g., in [26]. We can find common features in user interface within the selling products domain. These features express typical tasks with information (showing the price information, purchasing possibility, and the product detail information). When implementing a web site the web designers proceed the same way. They also use patterns even if they do not call them so (see [6]). During the analysis of selling products domain and even our experiments we worked with a ten of patterns, which come out from [6]. Figure 1 demonstrates an example where there is a cut of a web page with selling of product on eBay.com.

Fig. 1. A web page with marked patterns. The patterns found are graphically marked on the page: *Sign on possibility*, *Price information*, *Purchase possibility*, and *Rating*.

GUI and domain patterns are designated for web designers and domain experts. They are written with free text whereas structure of their description is formalized. For our purposes we do need to find a description of patterns which will be independent of the web designers and implementation and which will be useful for a semantic analysis of web pages.

Patterns on the page represent, in certain degree, what the user can expect (as a consequent of agreement between web designers and users). For us this is the key prerequisite for technical usage of patterns. Our algorithms are able to determine whether pattern is on the page or not and as a consequence of this we can annotate this page and use this annotation next time. We have to also answer the question what can this bring to the user.

Obviously, there is a problem in formalization of this approach. Patterns on the page do not appear in exact form. A crucial feature of presence of the pattern on the page is that individual elements of the pattern appear more or less together. More formal description of this deduction is described again in [24], [19]. The visual systems usually implement so-called Gestalt principles:

- *Proximity* – related information tends to be close to each other.
- *Similarity* – similarly looking elements contain similar information.
- *Continuity* – the layout of the information is continuous.
- *Closure* – related information tends to be enclosed.

So we can suppose a web pattern as a group of characteristic technical elements (which are based on GUI patterns) and a group of domain specific elements for the domain we are involved in (typical keywords related to given pattern and other entities such as the price, date, percent, etc.).

Example. On the right bottom of Figure 1 it is clearly visible that there is instance of *Rating* pattern.

- For technical implementation of this item it is used *div layout* which is useful for good information structuring.
- In the pattern instance we can find characteristic words – *seller*, *feedback*, *score*, *positive*.
- Moreover we can find integer, date and percentage data type instances – *17*, *90.5%* and *Oct-20-04*.

4 Query Simplifying

Patterns should give us an understandable language, which we are able to use in communication with the user to settle what he expects on the page (see Figure 2). Using search engines he has to think about a set of key words, e.g. “price”, which he uses to specify his requirement. The pattern *Price information* contains much stronger information that “price” occurring on the page.

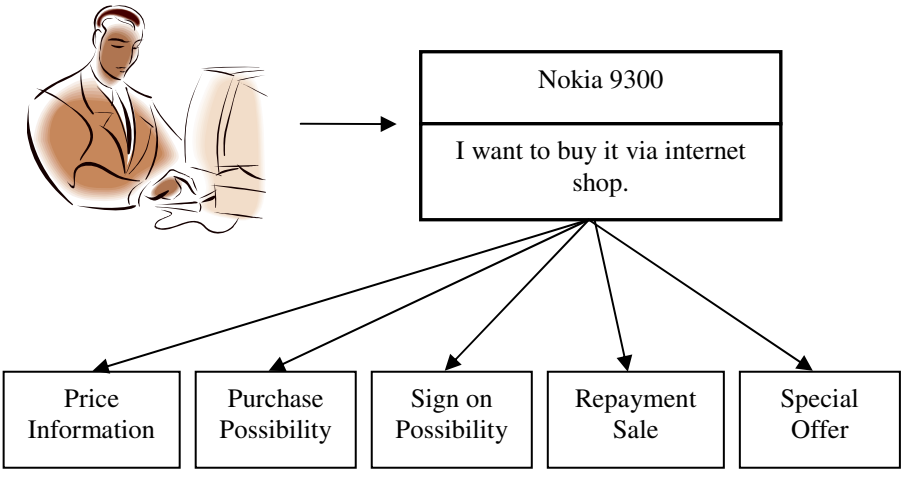


Fig. 2. Query from selling product domain. With the patterns we can set up catalogues and profiles which simplify the selection process for the user. It will remain to the user that he will have to enter subject of search but patterns will help him to specify expectations.

5 Improvement of Answer Relevance

When annotating pages we are working with the same information, which the user uses to make query. Moreover this information has stronger semantic content than, e.g., enumeration of keywords. Assume that we have annotated pages in a database with regard to patterns so there is information about which patterns are contained on each page. We can then use this information in two manners:

1. When showing web search engine results, for every shown page link in the returned set of links, we can add information about which patterns have been founded on the page. The user can recognize on the first look whether the page will fulfil his expectation.
2. We can count on the patterns already when performing search and sort page links with regard to weights of required patterns we have found on the page.

A side-effect in this situation is that there will be preferred those pages which have been created by high-quality web designers using recommended techniques described in patterns. Since patterns describe widely accepted solutions to users, a user will be given high-quality designed pages earlier in selection, i.e. on the first positions of the result set of page links.

6 Experiments Preparation

The key aspect of the pattern manifestation is that the introduced elements are close to each other. We can focus on the structure of the page during page analysis. It is not necessary to provide the deep analysis of page structure, because the technical elements provide just the environment for keeping the information together.

6.1 Choosing Domain and Domain Patterns

We consider the domain of e-commerce for testing purposes. Our goal was to integrate common users to testing so we have chosen one of the most used domains ever. During the domain analysis phase we sorted out nine domain web patterns – *Sign on possibility*, *Price information*, *Purchase possibility*, *Special Offer*, *Annuity selling*, *Product details*, *Comments and reviews*, *Discussion and FAQ*, *Advertising*. They are the semantic elements, whose are commonly expected by the user along the identification of the product.

6.2 Pattern Dictionary Preparation

For every one of the introduced web patterns we have manually chosen a group of words, which occurrence is characteristic for pattern on the web page. The words have been put into the database and serve as input for algorithms performing analysis of web pages. The set of chosen words needs to be understood as a starting set, which is automatically expanded according to deeper analysis of the pattern (we measure a frequency of words inside text segments of found patterns – the similar approach is presented in [3], [14]). Every pattern has its own dictionary of words.

We chose the words that are usually used by users when querying in the search engines – *price*, *eur*, *usd*, *offer*, *discount*, *stock*, *basket*, *buy*, *shop*, *review*, *forum*, *for sell*, *specifications*, … We have assigned more than five words to each pattern preferring most frequently used words in various associations.

These words can have multiple meanings and usually only one of the meanings is meant to represent a pattern. It is very important that the words in dictionaries are domain associated. We can then expect that [13]:

- The dictionary is not too large.
- The words occur in certain schemata.
- The meaning of words is more or less unambiguous.
- The words appear frequently in the text.

6.3 Patterns Extraction

The key task for solution of the problem we are dealing with is to find the mechanism of pattern detection on the page. We had to develop algorithms working with content of web page. They try to answer the questions about a weight of pattern on the page. In semantic analysis we need to find characteristics, which are

- dependent on the meaning of what a pattern represents for user, and
- independent of pattern implementation.

In our experiments we simplified pattern formalization problem using a set of words and data entities (e.g. date or percent), which are characteristic for the pattern. We can choose one concrete pattern (e.g. *Discussion*) and make analysis of big amount of web pages with discussions. After the analysis we can find out that there is quite small group of word and data entities by which it is possible to recognize the pattern. So if we suppose that we know the terminology for discussions (terms like *discussion*, *author*, *re*, etc.), then we can find segments in the plain text of the web page where the terms occur.

Let E be a set containing all *entities* they are characteristic for a given pattern (*pattern dictionary* – keywords and data types). On power set $P(E)$ we can define a binary relation δ so that pair (E, δ) makes a proximity space (see, e.g., [20]). Proximity space is used as the closeness model for groups of pattern entities. This defined structure is used as instrument for description and finding of page *text segments* S_i, which can (or does not have to) be a part of the given pattern. Let I be an instance of the given pattern. Then $I = \{S_1, ..., S_m\}$, where $m > 0$ and $S_i \in P(E)$.

A *pattern instance* is a set of analyzed text segments, which contain pattern entities (we do not focus on meaning of group of words but only on their presence). For discovery of algorithms that can be useful for finding and analyzing selected segments the Gestalt principles can do us a good turn. We developed methods for

- *proximity*: how to measure closeness (distance) between entities in searched text segments. We work with tree organization of entities representing a text segment and we suppose that in searched text segment entities must be close enough to each other (we have designated the distance based on analysis of text segments in searched pages).
- *similarity*: for measuring similarity of two searched text segments (for *Discussion* we are able to identify repetition of replies). We work with comparison of trees representing text segments.
- *continuity*: how to find out whether two or more found text segments make together instance of a pattern. We assume that two or more similar text segments (trees of entities from one pattern) match together.
- *closure*: for computation of the weight of one single searched text segment. In principle, we used two criteria. We rated shape of the segment tree (particularly, ratio of height and entity count) and quantity of all words and paragraphs in the text segment. On the overall computation of the weight also the proximity rate participates.

With complex usage of all mentioned principles we have implemented an algorithm, which offers excellent results in pattern extraction. The algorithm uses only plain text and regardless of the fact it is successful in more than 80% of cases.

6.4 Algorithm

Our algorithm is built on application of Gestalt principles. Before the algorithm takes place the page HTML code is preprocessed – the plain text (sequence of words) is extracted and data type instances (data entity) are found. The data type instances and sequence of words make list of page entities. Pattern dictionary is composed of characteristic words and expected data types (pattern entities).

Input for the algorithm is both set of entities, which represents each word, and data entity from the text of a web page and set of characteristic pattern entities. The algorithm compares these entities with characteristic pattern entities and creates representation of text segments called snippets [7], which can belong to (or compose) a pattern. These representations are then used for further computations and the result is value representing the weight of pattern occurrence on the page.

```
FOREACH page entity in all page entities
   IF page entity is pattern entity THEN
      IF not exist snippet to add page entity to THEN
         create new snippet in list of snippets
      ENDIF
      add page entity to snippet
   ENDIF
ENDFOR
FOREACH snippet in list of snippets
   compute proximity of snippet
   compute closure of snippet
   compute value(proximity, closure) of snippet
   IF value is not good enough THEN
      remove snippet from list of snippets
   ENDIF
ENDFOR
compute similarity of list of snippets
compute continuity of list of snippets
compute value(similarity, continuity) of pattern
RETURN value
```

Example. Let's go back to Figure 1 and instance of *Rating* pattern. After preprocessing of page HTML code there is only list of page entities (pattern entities are emphasized with bold font):

seller *information* <*par*> ***feedback score <num>*** <*par*> ***positive feedback <perc>*** <*par*> *member since* ***<date>*** *in united states* <*par*> *read* ***feedback*** *comments* <*par*> *add to favourite sellers* <*par*> *view* ***seller*** *other items*

After extraction of segments and evaluation of *proximity* and *closure* criteria there are only two remaining segments which are potential parts of pattern (in the case that there is more segments found they will be taken into account).

seller *information* <*par*> ***feedback score <num>***

positive feedback <perc> <*par*> *member since* ***<date>***

The evaluation of *similarity* and *continuity* criteria is performed on the found segments. In our example the two segments are not considered similar but supplementary to each other. In our example the computed final probability of pattern presence is higher than 80%.

6.5 Experimental Application

To test our approach we have implemented web application, which uses Google Web API. The system queries Google for a set of few tens of pages, which correspond to the user's conditions. Then the application downloads the pages and extracts plain text from the HTML code. On the plain text there are performed analysis, patterns extraction, and evaluation of pattern's weights. Then the page is evaluated as a whole. Pages are then sorted according to the overall computed value.

We tested our extract algorithms on PC Intel Pentium M 1.6 GHz with installed Windows XP. We extracted 9 patterns on each page. The performance of algorithms was 100 pages in about 1 second. The average time of 1 pattern extraction was approximately 10^{-3} seconds.

7 Result Analysis

There were more than 200 searches of products tested (cellular phones, computers, components and peripheries, electronics, sport equipment, cosmetics, books, CDs, DVDs, etc.) in four profiles. We usually worked with first thirty of found pages; for dozens of products we tested the set of first one hundred found pages. For wide testing we selected a miscellaneous group of people, e.g. students of different types of schools and also people dealing with product selling on the Internet.

During experiments we collected 31,738 various web pages that we got from the Google search engine using queries on products. After the analysis we discovered that on the 11,038 web pages there was not any extracted pattern even though our queries were focused on pages containing these patterns (queries from our application contained groups of elaborative words). Even though we must count with queries on which it is not possible to find enough relevant answers and also with inaccuracy of our algorithms it has to be noted one interesting thing. In spite of very precise query to searching engines the user has to count with approximately one quarter up to one third of irrelevant web pages, which do not contain expected information.

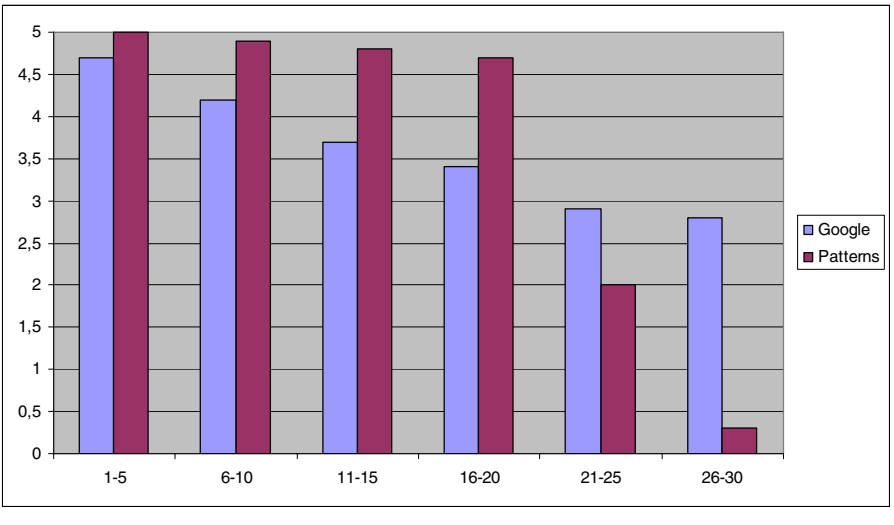


Fig. 3. First 30 pages (default and patterns sorted). On the horizontal axis, there are the first 30 found pages in groups of 5. On the vertical axis there is the number of relevant pages. The left columns show figures using Google search engine. Right columns show figures using our experimental application.

Figure 3 shows query results on concrete products. It includes only those queries on which there is multiply times more relevant pages than our first 30 analyzed web pages from search engine (it includes more than 200 various queries). We can observe that

- irrelevant web pages are moved to the end of the result set (our approach eliminates mistakes in order of pages),
- using our ranking the user reaches the expected pages earlier.

8 Future Work

Ability of inserting other web patterns into our system is opened. We are working on extraction of other web patterns from selling products domain (like *Rating*, *Ask the Seller*). It is also expected that the system will be extended with other domains in which it is able to identify web patterns. We are preparing *Tourism* and *Culture* domains.

During the query string analysis phase it is possible to recognize words which exists in our pattern dictionaries. Using this we are able to identify groups of patterns (called query profile) which user probably expects. We plan to use this observation during the result set links sorting and also during composing the user query expression.

We can see that patterns don't occur alone on pages but in certain groups. It means that there are page groups or clusters, which are characterized by the same web patterns. We can suppose such group of web patterns as a page profile. For searching page profiles we plan to use clustering methods.

9 Conclusion

The crucial aspect of our approach is that we do not need to analyze page's HTML code. Our algorithms are based on analysis of plain text of the page. For page evaluation we do not use any meta-information about page (such as title, hyperlinks, meta-tags, etc.). We also confirmed that key characteristics of web patterns are independent of language environment. We tested our method in English and Czech language environment. The only thing we had to do was to change patterns dictionaries.

Our experiments show that it is very useful to consider gained data about pattern existence as a metadata stored along with the page. So now we have tools, which are able to discover whether the page contains certain pattern with about 80% accuracy.

Our approach is not universal. The reason is that its basic assumption is a domain with relatively formed rules of how web pages look like (we do not expect uniformity but we expect some synchronization between users and web pages designers).

Acknowledgment. This research was supported in part by GACR grant 201/04/2102 and the National programme of research (Information society project 1ET100300419).

References

1. Baumgartner, R., Flesca, S., Gottlob, G.: Visual Web Information Extraction with Lixto. In: Proc. of the 27th Int. Conference on Very Large Data Bases, pp. 119–128 (2001)
2. Chakrabarti, S.: Mining the Web: Discovering Knowledge from Hypertext Data. Morgan Kaufman Publishers, San Francisco (2003)
3. Ciravegna, F., Chapman, S., Dingli, A., Wilks, Y.: Learning to harvest information for the semantic web. In: Bussler, C.J., Davies, J., Fensel, D., Studer, R. (eds.) ESWS 2004. LNCS, vol. 3053, pp. 312–326. Springer, Heidelberg (2004)
4. Cordón, O., Moya, F., Zarco, C.: Fuzzy Logic and Multiobjective Evolutionary Algorithms as Soft Computing Tools for Persistent Query Learning, in Text Retrieval Environments. In: IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE 2004), Budapest, Hungary, pp. 571–576 (2004)
5. Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., McCurley, K.S., Rajagopalan, S., Tomkins, A., Tomlin, J.A., Zien, J.Y.: A Case for Automated Large-Scale Semantic Annotation. Journal of Web Semantics 1(1), 115–132 (2003)
6. Van Duyne, D.K., Landay, J.A., Hong, J.I.: The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience. Pearson Education, London (2002)
7. Ferragin, P., Gulli, A.: A personalized search engine based on Web-snippet hierarchical clustering. In: Proc. of 14th Int. Conf. on World Wide Web, Chiba, Japan, pp. 801–810 (2005)
8. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns – Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1995)
9. Graham, I.: A pattern language for web usability. Addison-Wesley, Reading (2003)
10. Handschuh, S., Staab, S., Ciravegna, F.: S-CREAM – semi-automatic cREAtion of metadata. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS, vol. 2473, p. 358. Springer, Heidelberg (2002)
11. Husek, D., Owais, S., Kromer, P., Snasel, V., Neruda, R.: Implementing GP on Optimizing both Boolean and Extended Boolean Queries in IR and Fuzzy IR systems with Respect to the Users Profiles. In: 2006 IEEE World Congress on Computational Intelligence, CEC (accepted, 2006)
12. Ivory, M.Y., Megraw, R.: Evolution of Web Site Design Patterns. ACM Transactions on Information Systems 23(4), 463–497 (2005)
13. Jianming Li, L.Z., Yu, Y.: Learning to generate semantic annotation for domain specific sentences. In: Knowledge Markup And Semantic Annotation Workshop in K-CAP 2001 (2001)
14. Karov, Y., Edelman, S.: Similarity-based Word Sense Disambiguation. Computational Linguistics 24(1), 41–59 (1998)
15. Kiryakov, A., Popov, B., Ognyanoff, D., Manov, D., Kirilov, A., Goranov, M.: Semantic annotation, indexing, and retrieval. In: Fensel, D., Sycara, K.P., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 484–499. Springer, Heidelberg (2003)
16. Kiyavitskaya, N., Zeni, N., Cordy, J.R., Mich, L., Mylopoulos, J.: Semantic Annotation as Design Recovery. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729. Springer, Heidelberg (2005)
17. Kraft, D.H., Petry, F.E., Buckles, B.P., Sadasivan, T.: Genetic Algorithms for Query Optimization in Information Retrieval: Relevance Feedback. In: Sanchez, E., Shibata, T., Zadeh, L.A. (eds.) Genetic Algorithms and Fuzzy Logic Systems. World Scientific, Singapore (1997)

18. Li, Z., Ng, W.K., Sun, A.: Web data extraction based on structural similarity. Knowl. Inf. Syst. 8(4), 438–461 (2005)
19. Mullet, K., Sano, D.: Designing visual interfaces: Communication oriented techniques. Prentice-Hall, Englewood Cliffs (1994)
20. Naimpally, S.A., Warrack, B.D.: Proximity Spaces. Cambridge University Press, Cambridge (1970)
21. Pivk, A.: Automatic Ontology Generation from Web Tabular Structures. PhD thesis, University of Maribor (2005)
22. Reis, D.C., Golgher, P.B., Silva, A.S., Laender, A.F.: Automatic web news extraction using tree edit distance. In: WWW 2004: Proc. of the 13th Int. Conf. on World Wide Web, pp. 502–511. ACM Press, New York (2004)
23. Sean, L., Lee, S., Rager, D., Handler, J.: Ontology-based web agents. In: Proc. of the First Int. Conf. on Autonomous Agents (Agents 1997), pp. 59–68. ACM Press, New York (1997)
24. Tidwell, J.: Designing Interfaces: Patterns for Effective Interaction Design. O'Reilly Media, Inc., Sebastopol (2006)
25. Tijerino, Y.A., Embley, D.W., Lonsdale, D.W., Ding, Y., Nagy, G.: Towards Ontology Generation from Tables. World Wide Web 8(3), 261–285 (2005)
26. Wellhausen, T.: User Interface Design for Searching. A Pattern Language (May 29, 2005), http://www.timwellhausen.de/papers/UIForSearching/UIForSearching.html
27. Wechsler, K., Baier, J., Nussbaum, M., Baeza-Yates, R.: Semantic search in the WWW supported by a cognitive model. In: Li, Q., Wang, G., Feng, L. (eds.) WAIM 2004. LNCS, vol. 3129, pp. 315–324. Springer, Heidelberg (2004)

Towards Formal Interfaces for Web Services with Transactions*

Zhenbang Chen, Ji Wang, Wei Dong, and Zhichang Qi

National Laboratory for Parallel and Distributed Processing,
Changsha 410073, China
{z.b.chen,jiwang,dong.wei}@mail.edu.cn

Abstract. The accuracy of interface description is very important to service composition and dynamic selection of service-oriented systems. It is desirable to describe Web service formally so as to improve the ability of service orchestration. This paper presents a formal interface theory for specifying Web service by extending the existing with the ability to model interface behaviour with transactions at the levels of signature, conversation and protocol. Signature interface and conversation interface model the static invocation relations in Web service interfaces, and protocol interface describes the temporal invocation information. A formal semantics of protocol interface is presented. Based on the semantics, the protocol interface can be transformed into a Labeled Transition System (LTS). Additionally, the compatibility and substitutivity relation conditions between Web services are also proposed.

Keywords: Web service, Interface theory, Composition, Transaction.

1 Introduction

Service orientation is a new trend in software engineering [1]. It makes the separation of service provider and requester possible, and allows the run-time composition of services. Meanwhile, it is a challenge to understand and ensure a high confidence of service-oriented software systems.

Web service is emerging as a standard framework for service-oriented computing. Business integration raises the need for Web service composition languages such as BPEL4WS (BPEL) [2]. There are many works on formalization and verification for Web service composition languages, and their main aim is to ensure the correctness of Web service composition.

In service-oriented architecture, service providers publish the service interface descriptions to the service registry. It is important to ensure the interface accuracy for service-oriented computing. As a formal foundation of component-based design, de Alfaro and Henzinger [3] proposed a theory of interface automata for specifying the

* Supported by the National Natural Science Foundation of China under Grant No.60233020, 60303013, 90612009, 60673118; the National High-Tech Research and Development Plan of China under Grant No.2005AA113130; the National Grand Fundamental Research 973 Program of China under Grant No.2005CB321802; Program for New Century Excellent Talents in University under Grant No.NCET-04-0996.

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 292–304, 2009.

component interfaces. Recently, Beyer *et al.* [4] present a Web service interface description language, which can describe the Web service interfaces in three levels, i.e. signature, consistency and protocol. However, transactions are not considered in the existing interface theories, though it is one of the essential features in distributed computing such as Web service systems. How to describe transaction information of Web service is a problem. Web service-based transactions differ from traditional transactions in that they execute over long periods, require commitments to the transactions to be "negotiated" at runtime, and isolation levels must be relaxed [5].

The main contribution of this paper is to extend the formalism of Web service interfaces proposed in [4] for describing the transaction information in all three levels of signature, conversation and protocol. In each level, we separate the transaction description from the normal behaviour description. This separation makes the transaction information of Web service interfaces to be easier to describe and maintain. The compatibility and substitutivity relation conditions of the Web service interfaces are proposed for supporting Web service system development.

The rest of this paper is organized as follows. Section 2 presents the framework of the interface theory for Web services with transactions, including the signature interface, conversation interface and protocol interface. As a key element, Section 3 gives a complete semantics of protocol interface. Section 4 exemplifies the theory by a case study. In Section 5, related work is reviewed and compared with ours. Section 6 concludes the paper and presents some future work.

2 Web Service Interface Theory

A Web service interface description contains some method declarations, and clients can use the functionalities of Web services through method calls. A Web service may provide or request some methods which may return some different values. An *action* is one case of a method call. From the perspective of *action*, the interface behaviour of Web service contains *three parts*. The *first part* is the normal behaviour of action invocations. If an exception action is invoked and completes, it will be handled by its corresponding fault handling behaviour, which is the *second part* of the interface behaviour. If an exception action can invoke some successful actions before the exception occurrence, the successful actions should be compensated by the corresponding compensation behaviour, which is the *third part* of the interface behaviour.

There are different detailed interface descriptions from Web service providers. For this reason, we propose the interface theory for describing the transaction information at three different abstract levels of signature, conversation and protocol. Inspired by the ideas of Aspect-Oriented Programming (AOP) [6], we separate the descriptions of fault handling and compensation behaviour from those of normal behaviour in the interface description. In the interface semantics, the fault handling and compensation behaviour can be weaved with the normal behaviour to describe the transaction information.

Let $\mathcal{M}$ be a finite set of Web methods, $\mathcal{O}$ be a finite set of outcomes, and $dom(f)$ denote the domain of the function f. Each level of the interface theory will be presented as follows.

Definition 1 (Signature Interface, SI). *A signature interface* $\mathcal{P} = (\mathcal{A}, \mathcal{S}, \mathcal{S}_\mathcal{C}, \mathcal{S}_\mathcal{F})$*, where* $\mathcal{A} \subseteq \mathcal{M} \times \mathcal{O}$ *is a set of actions that can appear in* $\mathcal{P}$*,* $\mathcal{S} : \mathcal{A} \to 2^\mathcal{A}$ *is a partial function that assigns to an action* a *a set of actions that can be invoked by* a*,* $\mathcal{S}_\mathcal{C} : \mathcal{A} \to 2^\mathcal{A}$ *is a partial function that assigns to an action* a *a set of actions that can be invoked by the compensation for* a*,* $\mathcal{S}_\mathcal{F} : \mathcal{A} \to 2^\mathcal{A}$ *is a partial function that assigns to an action* a *a set of actions that can be invoked by the fault handling for* a*, and* $dom(\mathcal{S}_\mathcal{C}) \cap dom(\mathcal{S}_\mathcal{F}) = \emptyset$*,* $dom(\mathcal{S}_\mathcal{C}) \cup dom(\mathcal{S}_\mathcal{F}) = dom(\mathcal{S})$.

Signature interface describes the direct invocation relation of Web service interfaces. An action may have different *types*. An action $a \in \mathcal{A}$ is a *supported action* if $\mathcal{S}(a)$ is defined. A Web method $m \in \mathcal{M}$ is a *supported method* if there exists a supported action $a = \langle m, o \rangle$. An action a is a *success action* if $\mathcal{S}_\mathcal{C}(a)$ is defined. An action a is an *exception action* if $\mathcal{S}_\mathcal{F}(a)$ is defined. An action a is a *required action* if it can be invoked by a supported action or compensation or fault handling, which can be expressed by the formula defined as follows:

$$required(a') = (\exists a \in dom(\mathcal{S}).\, a' \in \mathcal{S}(a)) \vee (\exists a \in dom(\mathcal{S}_\mathcal{C}).\, a' \in \mathcal{S}_\mathcal{C}(a)) \vee (\exists a \in dom(\mathcal{S}_\mathcal{F}).\, a' \in \mathcal{S}_\mathcal{F}(a)).$$

Service registries often require service providers to publish solid interface descriptions. Well-formedness can be used to assure the integrity. A signature interface is *well-formed* if the following conditions hold: every required action whose method is a supported method is a supported action, and no exception action can be invoked in compensation or fault handling.

Given two Web service interfaces, it is desirable to check whether they can cooperate properly. First, two Web services cannot support the same actions. Second, the new Web service interface, which is composed of them, should be well-formed. Formally, given two signature interfaces $\mathcal{P}_1 = (\mathcal{A}_1, \mathcal{S}_1, \mathcal{S}_{\mathcal{C}1}, \mathcal{S}_{\mathcal{F}1})$ and $\mathcal{P}_2 = (\mathcal{A}_2, \mathcal{S}_2, \mathcal{S}_{\mathcal{C}2}, \mathcal{S}_{\mathcal{F}2})$, they are *compatible* (denoted by $comp(\mathcal{P}_1, \mathcal{P}_2)$) if the following conditions are satisfied: $dom(\mathcal{S}_1) \cap dom(\mathcal{S}_2) = \emptyset$, and $\mathcal{P}_c = \mathcal{P}_1 \cup \mathcal{P}_2 = (\mathcal{A}_1 \cup \mathcal{A}_2, \mathcal{S}_1 \cup \mathcal{S}_2, \mathcal{S}_{\mathcal{C}1} \cup \mathcal{S}_{\mathcal{C}2}, \mathcal{S}_{\mathcal{F}1} \cup \mathcal{S}_{\mathcal{F}2})$ is well-formed. If two signature interfaces $\mathcal{P}_1$ and $\mathcal{P}_2$ are compatible, their composition (denoted by $\mathcal{P}_1 \parallel \mathcal{P}_2$) is $\mathcal{P}_c$. The composition operator is commutative and associative.

To enable top-down design, it is desirable to replace a Web service in a system (environment) with a new Web service without affecting the running of the system. After replacement, all parts of the system can still cooperate properly as before. Intuitively, the supported, success and exception actions are the *guarantees* of the Web service, and the required actions are the *assumptions* of the environment. The replacing Web service should guarantee more and assume fewer than the replaced Web service.

Given two signature interfaces $\mathcal{P}_1 = (\mathcal{A}_1, \mathcal{S}_1, \mathcal{S}_{\mathcal{C}1}, \mathcal{S}_{\mathcal{F}1})$ and $\mathcal{P}_2 = (\mathcal{A}_2, \mathcal{S}_2, \mathcal{S}_{\mathcal{C}2}, \mathcal{S}_{\mathcal{F}2})$, $\mathcal{P}_2$ *refines* $\mathcal{P}_1$ ($\mathcal{P}_2 \preccurlyeq \mathcal{P}_1$) if the following conditions are satisfied:for every $a \in \mathcal{A}$, if $\mathcal{P}_1$ supports a, then $\mathcal{P}_2$ supports a; for every $a \in \mathcal{A}$, if a is a success action in $\mathcal{P}_1$, then a is a success action in $\mathcal{P}_2$; for every $a \in \mathcal{A}$, if a is an exception action in $\mathcal{P}_1$, then a is an exception action in $\mathcal{P}_2$; for every $a, a' \in \mathcal{A}$, and $a \in dom(\mathcal{S}_1)$, if $a' \in \zeta_2(a)$, where $\zeta_2 \in \{\mathcal{S}_2, \mathcal{S}_{\mathcal{C}2}, \mathcal{S}_{\mathcal{F}2}\}$, then $a' \in \zeta_1(a)$; for every unsupported Web method $m \in \mathcal{M}$ in $\mathcal{P}_2$, if $\langle m, o \rangle$ is a required action in $\mathcal{P}_2$, then $\langle m, o \rangle$ is a required action in $\mathcal{P}_1$.

The first three conditions ensure that the replacing Web service guarantees every action guaranteed by the replaced one. The last two conditions ensure that every required action in $\mathcal{P}_2$ is required by $\mathcal{P}_1$, and they describe that $\mathcal{P}_2$ assumes fewer actions which

are supported by environment than $\mathcal{P}_1$. Given three signature interfaces $\mathcal{P}_1$, $\mathcal{P}_2$, and $\mathcal{P}_3$, if $comp(\mathcal{P}_1, \mathcal{P}_3)$, $comp(\mathcal{P}_2, \mathcal{P}_3)$, and $\mathcal{P}_2 \preccurlyeq \mathcal{P}_1$, then $\mathcal{P}_2 \parallel \mathcal{P}_3 \preccurlyeq \mathcal{P}_1 \parallel \mathcal{P}_3$.

An action may invoke different action sets in different conditions. Signature interface cannot describe this feature. Conversation interface is proposed for specifying different cases of action invocation. A conversation is a set of actions that are invoked together. Propositional formulae are used to represent different conversations. The set of *conversation expressions* over an action set $\mathcal{A}$ is given by the following grammar, where $a \in \mathcal{A}$.

$$\omega :: \top \mid a \mid \omega_1 \sqcup \omega_2 \mid \omega_1 \sqcap \omega_2$$

$\top$ is the propositional constant, which represents no action is needed to be invoked. Action a represents a single action is needed to be invoked. The expression $\omega_1 \sqcup \omega_2$ represents that each conversation represented by ω_1 or ω_2 can be invoked. The expression $\omega_1 \sqcap \omega_2$ represents that one conversation must contain each conversation from ω_1 and ω_2. The set of all conversation expressions on the action set $\mathcal{A}$ is denoted by $\omega(\mathcal{A})$.

A conversation interface $\mathcal{I}$ = ($\mathcal{A}$, $\mathcal{E}$, $\mathcal{E}_\mathcal{C}$, $\mathcal{E}_\mathcal{F}$), where $\mathcal{A} \subseteq \mathcal{M} \times \mathcal{O}$ is a set of actions that can appear in $\mathcal{I}$, $\mathcal{E}$, $\mathcal{E}_\mathcal{C}$ and $\mathcal{E}_\mathcal{F}$ are partial functions, whose definitions are same as $\mathcal{A} \rightarrow \omega(\mathcal{A})$. The meanings of functions are similar to those of signature interface, except that each function assigns to an action a conversation expression to describe the interface behaviour.

A conversation is a set of actions, which do not have any sequence information. [4] presents *protocol interface* to depict the sequences of actions. In this paper, the protocol interface is extended to enable transaction description. In Web service, the modes of action invocations include thread creation, choice, parallel executions, join after parallel execution, sequence, *etc.* We use terms to represent these different modes. The set of terms over an action set $\mathcal{A}$ is given by the following grammar, where $a,\ b \in \mathcal{A}$.

$$term :: \tau \mid a \mid a \sqcup b \mid a \sqcap b \mid a \boxplus b \mid a\ ;\ b \mid [term]$$

The set of all terms over $\mathcal{A}$ is denoted by $Term(\mathcal{A})$, and $[[term]] = [term]$. The term τ is an *Empty term*, which represents that no action is invoked. The term a=$\langle m, o \rangle$ is a *Call term*, which represents a call to Web method m with expected outcome o. The term $a \sqcup b$ is a *Choice term*, which represents a nondeterministic choice between actions a and b. The term $a \sqcap b$ is a *Fork term*, which represents parallel invocations of actions a and b, and the parent waits for both actions to return. If any action is an exception action, the parent fails. The term $a \boxplus b$ is a *Fork-Choice term*, which represents parallel invocations of actions a and b, while the return of any action will return the parent. $a \boxplus b$ fails only when both actions are exception actions. The term $a\ ;\ b$ is a *Sequence term*, which represents two sequential calls. The term $[term]$ is a *Transaction term*, which represents that any exception action invoked from the term in the brackets will cause the compensation or fault handling to the actions which are invoked before from the the term in the brackets. For the sake of simplicity, it is assumed that the term of transaction term must result in exceptions.

The sequences of invocations between Web services can be specified in automata. To indicate the place where exceptions occur, we propose *extended protocol automata* as follows.

Definition 2 (Extended Protocol Automata, EPA). *An extended protocol automaton G is a triple $(\mathcal{A}, \mathcal{L}, \delta)$, where $\mathcal{A}$ is a set of actions, $\mathcal{L}$ is a set of locations, there are two special locations $\perp, \boxtimes$ in $\mathcal{L}$, $\perp \in \mathcal{L}$ is the return location, and $\boxtimes \in \mathcal{L}$ is the exception location, $\delta \subseteq (\mathcal{L} \setminus \{\perp, \boxtimes\}) \times Terms(\mathcal{A}) \times \mathcal{L}$ is the transition relation set.*

A location is terminating in *EPA* if there exists a trace starting from the location and ending with $\perp$ or $\boxtimes$. Based on *EPA*, we define protocol interface as follows.

Definition 3. *[Protocol Interface, PI] A protocol interface $\mathcal{T}$ is 4-tuple $(G, \mathcal{R}, \mathcal{R}_\mathcal{C}, \mathcal{R}_\mathcal{F})$, where G is an extended protocol automaton to specify interface behaviour, $\mathcal{R} : \mathcal{A} \to \mathcal{L}$ is a partial function that assigns to an action the start location in G, $\mathcal{R}_\mathcal{C} : \mathcal{A} \to \mathcal{L}$ is a partial function that assigns to an action a the start location in G of the compensation for a, $\mathcal{R}_\mathcal{F} : \mathcal{A} \to \mathcal{L}$ is a partial function that assigns to an action a the start location in G of the fault handling for a, and $dom(\mathcal{R}_\mathcal{C}) \cap dom(\mathcal{R}_\mathcal{F}) = \emptyset$, $dom(\mathcal{R}_\mathcal{C}) \cup dom(\mathcal{R}_\mathcal{F}) = dom(\mathcal{R})$.*

A location is terminating in *PI* if it is terminating in *EPA* and the start location of each action in the terminating trace is also terminating in *PI*. Given a protocol interface $\mathcal{T} = (G, \mathcal{R}, \mathcal{R}_\mathcal{C}, \mathcal{R}_\mathcal{F})$, the *underlying signature interface* of $\mathcal{T}$ (denoted by $psi(\mathcal{T})$) is $(\mathcal{A}_s, \mathcal{S}, \mathcal{S}_\mathcal{C}, \mathcal{S}_\mathcal{F})$, where $\mathcal{A}_s = \mathcal{A}$; $\mathcal{S}(a) = sigl(\mathcal{R}(a))$ if $\mathcal{R}(a)$ is defined, otherwise $\mathcal{S}(a)$ is undefined; $\mathcal{S}_\mathcal{C}(a) = sigl(\mathcal{R}_\mathcal{C}(a))$ if $\mathcal{R}_\mathcal{C}(a)$ is defined, otherwise $\mathcal{S}_\mathcal{C}(a)$ is undefined; $\mathcal{S}_\mathcal{F}(a) = sigl(\mathcal{R}_\mathcal{F}(a))$ if $\mathcal{R}_\mathcal{F}(a)$ is defined, otherwise $\mathcal{S}_\mathcal{F}(a)$ is undefined. The function $sigl : \mathcal{L} \to 2^\mathcal{A}$ is defined as follows:

$sigl(\perp) = \emptyset, sigl(\boxtimes) = \emptyset, sigl(q) = \bigcup_{\exists(q,term,q') \in \delta} \varphi(term) \cup sigl(q')$,

$\varphi(\tau) = \emptyset, \varphi(a) = \{a\}, \varphi([term]) = \varphi(term), \varphi(a \square b) = \{a, b\}, \square \in \{\sqcup, \sqcap, \boxplus, ;\}$.

A protocol interface $\mathcal{T}$ is *well-formed* if the following conditions hold: $psi(\mathcal{T})$ is well-formed; if $a \in dom(\mathcal{R})$, then $\mathcal{R}(a)$ is terminating; if $a \in dom(\mathcal{R}_\mathcal{C})$, then $\mathcal{R}_\mathcal{C}(a)$ is terminating; if $a \in dom(\mathcal{R}_\mathcal{F})$, then $\mathcal{R}_\mathcal{F}(a)$ is terminating. For the sake of simplicity, it is assumed that: no transaction term can be invoked by exception action, nor can it be invoked by compensation or fault handling; transaction term cannot be invoked recursively or parallelly. The types of an action a in a protocol interface $\mathcal{T}$ are same as those of a in $psi(\mathcal{T})$.

Given two protocol interfaces $\mathcal{T}_1 = (G_1, \mathcal{R}_1, \mathcal{R}_{\mathcal{C}1}, \mathcal{R}_{\mathcal{F}1})$ and $\mathcal{T}_2 = (G_2, \mathcal{R}_2, \mathcal{R}_{\mathcal{C}2}, \mathcal{R}_{\mathcal{F}2})$, they are *compatible* if the following conditions are satisfied: $psi(\mathcal{T}_1)$ and $psi(\mathcal{T}_2)$ are compatible, and $\mathcal{L}_1 \cap \mathcal{L}_2 = \{\perp, \boxtimes\}$; $\mathcal{T}_c = \mathcal{T}_1 \cup \mathcal{T}_2 = (G_1 \cup G_2, \mathcal{R}_1 \cup \mathcal{R}_2, \mathcal{R}_{\mathcal{C}1} \cup \mathcal{R}_{\mathcal{C}2}, \mathcal{R}_{\mathcal{F}1} \cup \mathcal{R}_{\mathcal{F}2})$ is well-formed, where $G_1 \cup G_2 = (\mathcal{A}_1 \cup \mathcal{A}_2, \mathcal{L}_1 \cup \mathcal{L}_2, \delta_1 \cup \delta_2)$. If $\mathcal{T}_1$ and $\mathcal{T}_2$ are compatible (denoted by $comp(\mathcal{T}_1, \mathcal{T}_2)$), their composition (denoted by $\mathcal{T}_1 \parallel \mathcal{T}_2$) is $\mathcal{T}_c$. The composition operator is commutative and associative. The substitutivity relation between protocol interfaces should be defined based on the semantics to ensure the temporal correctness.

Signature interface and conversation interface describe the static invocation relations of Web service interfaces, and their semantics are simple and show the static aspects of the Web service interface. Protocol interface describes dynamic invocations in Web service interfaces. The interface behaviour of protocol interface should ensure not only the invocation process should been recorded for compensation or fault handing, but also the sequence of compensation and fault handling should agree to the long-running transaction model.

3 The Semantics of Protocol Interface

The action invocation process is a *pushdown* system which can continue only after the completion of every invoked action. The sequence of compensation and fault handling should be reverse of the sequence of the previous invocations, and the recorded actions should be first in last out. Therefore, we use a binary tree nested by a stack to interpret the protocol interface behaviour.

A *binary tree* over a finite set of labels L is a partial function $t : \mathbb{B}^* \rightarrow L$, where $\mathbb{B}^*$ denotes the set of finite words over $\mathbb{B} = \{0, 1\}$, and ρ denotes the empty word. $\mathbb{T}(L)$ denotes the set of all trees over a finite label set L. A *stack* over a finite set of labels L is a partial function $s(m) : \mathbb{N} \rightarrow L$, where $\mathbb{N}$ is the natural number set, and $dom(s(m)) = \{n \mid n < m \wedge n \in \mathbb{N}\}$. $s(0)$ is the empty stack. $s(m)(m-1)$ is the top element of the stack $s(m)$. $\mathbb{S}(L)$ denotes the set of all stacks over a finite label set L.

Given a protocol interface $(G, \mathcal{R}, \mathcal{R}_\mathcal{C}, \mathcal{R}_\mathcal{F})$, its semantics is defined by a *labeled transition system* (LTS). The set of states is $\mathbb{T}(Q_t) \times \mathbb{B}^* \times \mathbb{S}(Term(\mathcal{A}))$, that is, the Cartesian products of trees over $Q_t = \mathcal{L} \times \mathcal{A}^* \times \wp$, the set of tree nodes $\mathbb{B}^*$, and stacks over $Term(\mathcal{A})$, where $\mathcal{L}$ is the location set of $EPA\ G$, $\mathcal{A}^*$ is the set of words over the action set $\mathcal{A}$ in G, $\wp = \{\circ, \boxdot, \boxdot_c, \boxdot_d, \boxplus, \boxplus_c, \boxplus_d, \triangle, \nabla, \odot, \boxdot\}$ is the node type set, and the tree node in $\mathbb{B}^*$ is the corresponding node of the stack. The *underlying transition relation* of $\mathcal{T}$ is a transition relation $\rightarrow_\mathcal{T} \subseteq (\mathbb{T}(Q_t) \times \mathbb{B}^* \times \mathbb{S}(Term(\mathcal{A}))) \times 2^{\mathcal{A} \cup \{ret, exp, cfstart, end\}} \times (\mathbb{T}(Q_t) \times \mathbb{B}^* \times \mathbb{S}(Term(\mathcal{A})))$. The label of state transition is the set of elements from $\mathcal{A} \cup \{ret, exp, cfstart, end\}$. We write $\nu \rightarrow_\mathcal{B} \nu'$ for $(\nu, \mathcal{B}, \nu') \in \rightarrow_\mathcal{T}$, where $\nu = (t, \Psi, s(n))$ and $\mathcal{B} \subseteq \mathcal{A} \cup \{ret, exp, cfstart, end\}$. The transition rules have two parts, the first part consists of the rules for normal behaviour transitions, and the second part consists of the rules for transitions made by compensation or fault handling. The second part weaves the transaction behaviour into the normal behaviour.

The beginning of normal behaviour is the invocation of a supported action. The initial state is a tree that has only one node and a stack whose content is decided by the type of the supported action. Supposing we start from invoking a supported action a, if a is a success action, the initial state is $\nu_{inital} = (t_{inital}, \Psi, s(0)) = (\{(\rho, (\mathcal{R}(a), \rho, \circ))\}, \rho, \emptyset)$, else if a is an exception action, the initial state is $\nu_{inital} = (t_{inital}, \Psi, s(1)) = (\{(\rho, (\mathcal{R}(a), \rho, \circ))\}, \rho, \{(0, a)\})$. The operations in normal transaction rules can be divided into two parts: *tree operations* and *stack operations*. The tree operations depict a pushdown system. Only leaf nodes of the tree can be operated. Call, Choice and Sequence terms lead to pushing down the leaf node. Fork and Fork-Choice terms lead to branching the leaf node. Transaction term leads to pushing down the leaf node with a transaction node. For pushdown operations, if current node location is reached from a success supported action, or the leaf node is not pushed from a transaction term node, no stack operations is needed. If current node location is reached from a supported exception action, or the node is branched from a transaction term node and the transaction node is not under compensation or fault handling, stack operations is needed. Main operations in the normal rules are illustrated in Figure 1.

There are five normal transition rules. As a shorthand, the rules (**Pushdown**) and (**Exception**) are listed in Appendix. The rule (**Pushdown**) describes the operations of different invoked terms. The rule (**Exception**) describes that when an exception location is reached, some coordination should be taken. There are two complicated

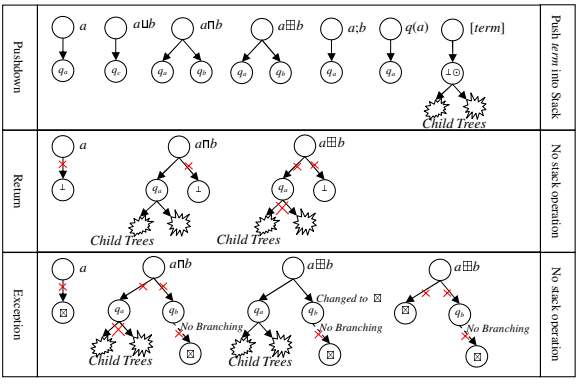


Fig. 1. Illustrations of operations in normal transition rules

cases. The first case is that the exception location is reached from a Fork term, and it will cause the global exception and the other branch should be terminated. Another case is that the exception location is reached from a Fork-Choice term, and whether it can cause the global exception is determined by the other branch. If the other branch returns successfully, the parent is successful. If the other branch does not return, this branch should wait until the return of the other branch. If exception also occurs in the other branch, the global exception occurs. The invocation of the unsupported action is supposed to return immediately.

The compensation and fault handling occur in the condition that some exception actions have been invoked. After normal transitions, actions in the execution must return, and the state of the execution must reach $(\{(\rho, (\perp, \rho, \circ))\}, \rho, s(0))$ or $(\{(\rho, (\boxtimes, \rho, \circ))\}, \rho, s(n))$ with $n > 0$. If it is $(\{(\rho, (\perp, \rho, \circ))\}, \rho, s(0))$, the invocation is successful, and no compensation or fault handling is needed. If the state is $(\{(\rho, (\boxtimes, \rho, \circ))\}, \rho, s(n))$, some exceptions occur in the invocation process, and compensation or fault handling should be taken. After compensation or fault handling, the state must finally reach $(\{(\rho, (\perp, \rho, \circ))\}, \rho, s(0))$, which represents that the whole invocation process completes. If the invocation returns on a node whose corresponding label is $(\boxtimes, \rho, \odot)$, $\Phi \neq \rho$, and the stack is not empty, it represents that a transaction term invocation has returned, and compensation or fault handling should be taken.

It is assumed that no exception will appear in compensation or fault handling. The compensation and fault handling do not need stack operation. The operations in compensation and fault handling transition rules are simpler than those in normal transition rules. The transition rules specify that the recorded term in the stack should be popped out in sequence, and whether compensation or fault handling will be taken is determined by the action type. After completing compensation and fault handling, whether the invocation process terminates is determined by the exception reason. If the beginning action is a supported exception action, the invocation process terminates. If the exception is resulted from a *Transaction term*, the invocation process will continue.

Based on the transition rules, we can use the *LTS* simulation relation to define the substitutivity relation of protocol interfaces. A labeled transition system is a 4-tuple (S, I, L, Δ), where S is the set of states, $I \subseteq S$ is the set of initial states, L is the set

of labels, and $\Delta \subseteq S \times L \times S$ is the transition relation set. Given a protocol interface $\mathcal{T} = (G, \mathcal{R}, \mathcal{R}_{\mathcal{C}}, \mathcal{R}_{\mathcal{F}})$, and a supported action a, the *underlying labeled transition system* of invoking a (denoted by $LTS(\mathcal{T}, a)$) is $(S_a, I_a, L_a, \Delta_a)$, which can be given as follows: $S_a = \mathbb{T}(Q_t) \times \mathbb{B}^* \times \mathbb{S}(Term(\mathcal{A}))$; if a is a success action, $I_a = \{(\{(\rho, (\mathcal{R}(a), \rho, \circ))\}, \rho, \emptyset)\}$; if a is an exception action, $I_a = \{(\{(\rho, (\mathcal{R}(a), \rho, \circ))\}, \rho, \{(0, a)\})\}$; $L_a = 2^{\mathcal{A} \cup \{ret, exp, cfstart, end\}}$; Δ_a is the underlying transition relation set of $\mathcal{T}$ using the transition rules.

Because ret, exp, $cfstart$, and end are not external Web service actions, the transitions labeled by them do not assume to the environment. The simulation relation of the underlying labeled transition systems can be extended to relax the conditions that substitutivity should satisfy. We denote $(s_1, a, s_1') \in \Delta$ as $s_1 \rightarrow_a s_1'$. If $t = a_1 a_2 ... a_n \in L^*$, $s_1 \rightarrow_{a_1} \rightarrow_{a_2} ... \rightarrow_{a_n} s_1'$ is denoted as $s_1 \rightarrow_t s_1'$.

Given two *LTSs* $M_1 = (S_1, I_1, L_1, \Delta_1)$ and $M_2 = (S_2, I_2, L_2, \Delta_2)$ and a label set W, M_2 is *weakly simulated* by M_1 over the label set W if there exists a relation $\preccurlyeq_W \subseteq S_1 \times S_2$ such that: for every $s_1 \in M_1, s_2 \in M_2$, if $s_2 \preccurlyeq_W s_1$, then for every $(s_2, a, s_2') \in \Delta_2$, there exists $s_1 \rightsquigarrow_a s_1'$ in Δ_1, such that $s_2' \preccurlyeq_W s_1'$, where $s_1 \rightsquigarrow_a s_1'$ represents $s_1 \rightarrow_t s_1'$, in which $t = a_1 a_2 ... a_n$, and there exists only one a_i that $a_i = a$, and all the other labels are all in W; for every $s_2 \in M_2$, there exists $s_1 \in M_1$ such that $s_2 \preccurlyeq_W s_1$.

Given two protocol interfaces $\mathcal{T}_1 = (G_1, \mathcal{R}_1, \mathcal{R}_{\mathcal{C}1}, \mathcal{R}_{\mathcal{F}1})$ and $\mathcal{T}_2 = (G_2, \mathcal{R}_2, \mathcal{R}_{\mathcal{C}2}, \mathcal{R}_{\mathcal{F}2})$, we say $\mathcal{T}_2$ *refines* $\mathcal{T}_1$ ($\mathcal{T}_2 \preccurlyeq \mathcal{T}_1$) if the following conditions are satisfied: $psi(\mathcal{T}_2) \preccurlyeq psi(\mathcal{T}_1)$, and for every $a \in dom(\mathcal{R}_1)$, $LTS(\mathcal{T}_2, a)$ is weakly simulated by $LTS(\mathcal{T}_1, a)$ over $2^{\{ret, exp, cfstart, end\}}$. Given three protocol interfaces $\mathcal{T}_1$, $\mathcal{T}_2$, and $\mathcal{T}_3$, if $comp(\mathcal{T}_1, \mathcal{T}_3)$, $comp(\mathcal{T}_2, \mathcal{T}_3)$, and $\mathcal{T}_2 \preccurlyeq \mathcal{T}_1$, then $\mathcal{T}_2 \parallel \mathcal{T}_3 \preccurlyeq \mathcal{T}_1 \parallel \mathcal{T}_3$.

4 Case Study

Figure 2 shows a classical Web service-based system, supply chain management system. The system is composed of six Web services. Each labeled arrow from one service to another indicates the Web method call from the caller to the callee. *Shop* supports the Web method *SellItem* that can be called by the *Client* to start the selling process. When the selling process starts, the *Shop* will first check the availability of items to be sold by calling the method *ChkAvail*, which requires the Web method *ChkStore* implemented by *Store* to check whether the desirable items are in stock and deduct the number of items if the stock checking is successful. If the stock is inadequate, the selling process fails. If the stock is inadequate or the stock after deducting is below a certain amount, the *Store* department will make an order from the *Supplier* and get some new items. If the availability checking is successful, the *Shop* will parallelly process the payment by calling the method *ProcPay* and the delivery by calling the method *ShipItem*. *ProcPay* is implemented by the *Bank* and its success can be compensated by calling the method *Compensate*. *ShipItem* is implemented by *Transport* and its success can be compensated by calling the method *Withdraw*. If all the above steps are successful, the selling process is successful, otherwise the successful steps before failure should be compensated and the failed steps should be handled. For instance, the *Shop* will call the method *Apologize* implemented by itself to send an apologetic letter to the *Client* because of the failure of the selling process. After composition, the supply chain management

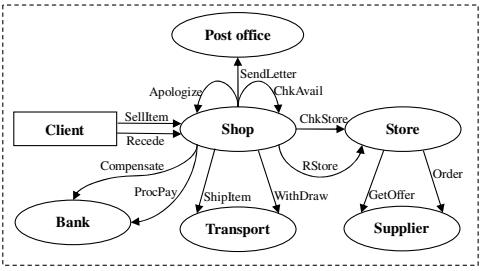


Fig. 2. Supply chain management system

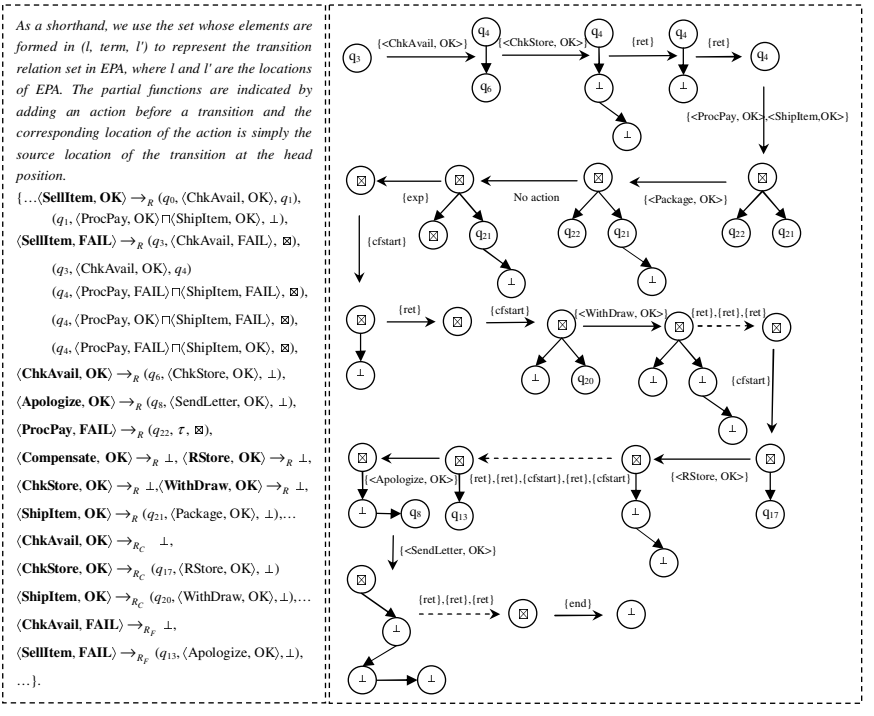


Fig. 3. The protocol interface for the supply chain management system and a trace of the action ⟨*SellItem,FAIL*⟩ invocation

system contains six Web services. The protocol interface for the system and a trace of the action ⟨SellItem, FAIL⟩ invocation is shown in Figure 3, and the reason of purchasing failure is the invocation of action ⟨ProcPay, FAIL⟩.

5 Related Work

There are many researches on formalization and verification of Web service interface behaviour, which can be divided by the underlying semantic theory as follows:

Petri net. [8] uses Petri net to formalize the Web service compositions. [9] uses WF-nets (workflow nets) to formalize the BPEL description, and a mapping from BPEL process model to WF-net was proposed.

Process algebra. Foster *et al.* [10] use finite state process (FSP) to formalize BPEL and WS-CDL [11], and the specifications are verified on LTSA WS-Engineer, which can perform safety and liveness analysis, and interface compatibility checking. In [12], BPEL-Calculus is proposed to formalize BPEL, and the description can be verified on Concurrency WorkBench (CWB) by a syntax compiler plug-in for BPEL-Calculus.

Automata. In [13], guarded finite state automata (GFSA) is used to describe service composition, and the description can be translated to Promela which can be verified on SPIN. In [14], hierarchical state machine (HSM) is used to specify Web service interfaces, and Java PathFinder (JPF) is used to verify that the implementation of each peer of Web service system conforms to its interface, and the interface behaviour models can be verified on SPIN.

The above approaches are deficient in modeling transaction behaviour of Web service interfaces, especially in compensation and fault handling. Because of the deficiency in formalism, the above verification methods cannot verify the properties which specify transaction behaviour part in Web service interfaces. By extending [4], the formalism presented in this paper can rigorously describe the transaction behaviour of Web service interfaces. The above verification methods mainly take into account the temporal interface behaviour and properties of Web services. In practice, some Web service providers cannot publish interfaces with temporal information. The approaches in [8,9,10,12,13,14] cannot handle this situation. Additionally, there are some researches on formalization of long-running transactions. Butler *et al.* [16] extend communicating sequential process (CSP) to enable the description of long-running transactions. In [17], an enhanced Sagas language is proposed for specifying compensation in flow composition languages. Their approaches mainly aim at description of dynamic transactional behaviour. Our approach takes into account different abstract levels and separates the transaction description from the normal behaviour description.

6 Conclusions

Web service is the most popular implementing framework for service-oriented computing. In this paper, we aim at formalization of Web service interfaces with transactions. The final goal is to ensure the correctness of Web service interfaces. This paper presents an interface theory for specifying interface behaviour with transactions in Web services. The signature interface specifies the direct invocation relations and the conversation interface specifies different invocations for the same call. With protocol interface, temporal invocations can be specified. To serve as a dynamic semantics of interfaces, a set of operational rules are presented to transform the protocol interface behaviour into labeled transition systems. The relation conditions of compatibility and substitutivity between Web services are also presented in this paper. The separate description method used in this paper reflects some ideas of AOP, and different parts of behaviour can be weaved together in the semantics.

Through our approach, one can precisely describe Web service interface transaction information. The interface theory for Web services will form as a formal foundation of service-oriented software engineering, especially in the specification and verification of service-oriented systems.

Currently, a model checking based verification method has been proposed [7] and our interface theory has been be applied to BPEL [18], and the application raises some issues which need improvements of the interface theory, such as data handling description. Other ongoing and future works are to investigate an integrated formalism for both service orchestration and choreography by Web service interface theory, and to build the corresponding tools for specification and verification.

References

1. Humberto, C., Richard, S.H.: Technical Concepts of Service Orientation. Service-Oriented Software System Engineering: Challenges and Practices, 1–47 (2005)
2. Curbera, F., et al.: Business Process Execution Language For Web Services Version 1.1, `ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf`
3. de Alfaro, L., Henzinger, T.A.: Interface automata. In: Proc. of ESEC/FSE 2001, pp. 109–120 (2001)
4. Beyer, D., Chakrabarti, A., Henzinger, T.A.: Web Service Interfaces. In: Proc. of WWW 2005, pp. 148–159. ACM Press, New York (2005)
5. Little, M.: Transactions and Web Services. Communication of the ACM 46(10), 49–54 (2003)
6. Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C.V., Loingtier, J.-M., Irwin, J.: Aspect-oriented programming. In: Aksit, M., Matsuoka, S. (eds.) ECOOP 1997. LNCS, vol. 1241, pp. 220–242. Springer, Heidelberg (1997)
7. Chen, Z.B., Wang, J., Dong, W., Qi, Z.C., Yeung, W.L.: An Interface Theory Based Approach to Verification of Web Services. In: Proc. of COMPSAC 2006, pp. 139–144. IEEE Press, Los Alamitos (2006)
8. Hamadi, R., Benatallah, B.: A Petri Net-based Model for Web Service Composition. In: Proc. of ADC 2003, pp. 191–200. IEEE Press, Los Alamitos (2003)
9. Verbeek, H.M.W., van der Aalst, W.M.P.: Analyzing BPEL Processes using Petri Nets. In: Proc. of the 2th International Workshop on Applications of Petri Nets to Coordination, Workflow and Business Process Management at the Petri Nets 2005, pp. 59–78 (2005)
10. Foster, H., Uchitel, S., Magee, J., Kramer, J.: Compatibility for Web Service Choreography. In: Proc. of ICWS 2004, pp. 738–741. IEEE Press, Los Alamitos (2004)
11. Kavantzas, N., et al.: Web Services Choreography Description Language Version 1.0, `http://www.w3.org/TR/ws-cdl-10/`
12. Koshkina, M.: Verification of Business Processes for Web Services. [MS. Thesis]. York University (2003)
13. Fu, X., Bultan, T., Su, J.W.: Analysis of Interacting BPEL Web Services. In: Proc. of WWW 2004, pp. 621–630. ACM Press, New York (2004)
14. Betin-Can, A., Bultan, T.: Verifiable Web Services with Hierarchical Interfaces. In: Proc. of ICWS 2005, pp. 85–94. IEEE Press, Los Alamitos (2005)
15. Beyer, D., Chakrabarti, A., Henzinger, T.A.: An Interface Formalism for Web Services. In: Abadi, M., de Alfaro, L. (eds.) CONCUR 2005. LNCS, vol. 3653. Springer, Heidelberg (2005)

16. Butler, M., Ripon, S.: Executable semantics for compensating CSP. In: Bravetti, M., Kloul, L., Zavattaro, G. (eds.) EPEW/WS-EM 2005. LNCS, vol. 3670, pp. 243–256. Springer, Heidelberg (2005)
17. Bruni, R., Melgratti, H., Montanari, U.: Theoretical Foundations for Compensations in Flow Composition Languages. In: Proc. of POPL 2005, pp. 209–220. ACM Press, New York (2005)
18. Chen, Z.B., Wang, J., Dong, W., Qi, Z.C.: Interface Theory based Formalization and Verification of Orchestration in BPEL4WS. International Journal of Business Process Integration and Management 2(4), 262–281 (2007)

Appendix. Some Important Normal Transition Rules

Due to the space limitations, we only give two normal transition rule definitions, where

- Let pj denote the concatenation of the word p with $j \in \mathbb{B}$, and pp' denote the concatenation of the words p and p'. For a tree t and a node $p \in dom(t)$, $leaf(t) = \{p \in dom(t) \mid \forall j \in \mathbb{B}, pj \notin dom(t)\}$, $child(p)=\{q \mid \exists j \in \mathbb{B}, q=pj \wedge q \in dom(t)\}$, and $parent(p) = \{q \mid \exists j \in \mathbb{B}, p = qj \wedge q \in dom(t)\}$. $ancestor(p_1, p_2) = (p_1 \in parent(p_2)) \vee (\exists q \in parent(p_2).\ ancestor(p_1, q))$ denotes whether a node p_1 is the ancestor of another node p_2. $ancestor\text{-}y(p, \beta)$ denotes the node p's youngest ancestor whose node type is β;
- in a transition, the source state is defined as $\nu = (t, \Psi, s(n))$, and the target state as $\nu' = (t', \Psi', s'(m))$. $q(w)\beta$ represents (q, w, β) in Q_t, and if $w = \rho$, $q\beta$ is used to represent it. For example, $q\boxplus$ represents $(q, \rho, \boxplus)$;
- $\delta(q) = (a, q')$ denotes that there exists a transition relation (q, a, q') in the extended protocol automaton. $\xi(p) = \square$ if the type of the tree node p is $\square \in \wp$. If action c is supported by $\mathcal{R}$, $q_c = \mathcal{R}(c)$, otherwise $q_c = \perp$.

(Pushdown) $\nu \rightarrow_{\mathcal{M}} \nu'$

If there exists a node p such that $p \in leaf(t)$, $t(p) = q\beta$, where $\beta \in \wp$, $\delta(q) = (r, q')$, and $\Psi = \rho \vee (\Psi \neq \rho \wedge \xi(\Psi) = \odot)$:

- $r = a$: $t' = (t \setminus \{(p, q\beta)\}) \cup \{(p, q'\beta), (p0, q_a\beta)\}$, $term_1 = a$, and $\mathcal{M} = \{a\}$;
- $r = a \sqcup b$: $t' = (t \setminus \{(p, q\beta)\}) \cup \{(p, q'\beta), (p0, q_c\beta)\}$, where $c \in \{a, b\}$, $term_1 = c$, and $\mathcal{M} = \{c\}$;
- $r = a \sqcap b$: $t' = (t \setminus \{(p, q\beta)\}) \cup \{(p, q'\alpha), (p0, q_a \circledcirc_c), (p1, q_b \circledcirc_c)\}$, where $(\beta = \circ \wedge \alpha = \circledcirc) \vee (\beta = \circledcirc_c \wedge \alpha = \circledcirc_d) \vee (\beta = \boxplus_c \wedge \alpha = \nabla)$, $term_1 = a \sqcap b$, and $\mathcal{M} = \{a, b\}$;
- $r = a \boxplus b$: $t' = (t \setminus \{(p, q\beta)\}) \cup \{(p, q'\alpha), (p0, q_a \boxplus_c), (p1, q_b \boxplus_c)\}$, where $(\beta = \circ \wedge \alpha = \boxplus) \vee (\beta = \boxplus_c \wedge \alpha = \boxplus_d) \vee (\beta = \circledcirc_c \wedge \alpha = \triangle)$, $term_1 = a \boxplus b$, and $\mathcal{M} = \{a, b\}$;
- $r = a; b$: $t' = (t \setminus \{(p, q\beta)\}) \cup \{(p, q'(b)\beta), (p0, q_a\beta)\}$, $term_1 = a$, and $\mathcal{M} = \{a\}$;
- if $t(p) = q(a)\beta$, where $\beta \in \wp$, then $t' = (t \setminus \{(p, q(a)\beta)\}) \cup \{(p, q\beta), (p0, q_a\beta)\}$, and $term_1 = a, \mathcal{M} = \{a\}$.

If $n = 0 \vee (\Psi \neq \rho \wedge p \neq \Psi p')$, then $s'(m) = s(n), m = n$. If $(n > 0) \wedge ((\Psi = \rho) \vee (\Psi \neq \rho \wedge p = \Psi p' \wedge \xi(\Psi) = \odot))$, then $m = n + 1$, and $s'(m) = s(n) \cup \{(n, term_1)\}$.

(Exception) $\nu \rightarrow_{\{exp\}} \nu'$

If there exists a node $p\theta$ such that $p\theta \in leaf(t)$, where $\theta \in \mathbb{B}$, $t(p\theta) = \boxtimes\beta$, where $\beta \in \wp$, and $\Psi = \rho \vee (\Psi \neq \rho \wedge \xi(\Psi) = \odot)$:

- $\beta = \circ$ and $t(p) = q\beta$: $t' = (t \setminus \{(p\theta, \boxtimes\circ), (p, q\beta)\}) \cup \{(p, \boxtimes\beta)\}$;
- $\beta = \circledcirc_c$, $t(p_a) = q\alpha$, where $\alpha \in \{\circledcirc, \circledcirc_d, \nabla, \odot\}$, and $ancestor\text{-}y(p\theta, \alpha) = p_a$: $t' = (t \setminus \{(p_a p', q') \mid p' \in \mathbb{B}^* \wedge q' = t(p_a p')\}) \cup \{(p_a, \boxtimes\beta)\}$, where $(\alpha = \circledcirc \wedge \beta = \circ) \vee (\alpha = \circledcirc_d \wedge \beta = \circledcirc_c) \vee (\alpha = \nabla \wedge \beta = \boxplus_c) \vee (\alpha = \odot \wedge \beta = \odot)$;
- $\beta = \boxplus_c$, and $t(p_a) = q\alpha$, where $\alpha \in \{\boxplus, \boxplus_d, \triangle, \odot\}$ and $ancestor\text{-}y(p\theta, \alpha) = p_a$: if there exists $p_b \in \{p' \mid child(p_a) \wedge \neg ancestor(p', p)\}$, and $\#child(p_b) > 0 \vee \xi(p_b) = \perp$, then $t' = (t \setminus \{(p_c p', q') \mid p' \in \mathbb{B}^* \wedge q' = t(p_c p')\}) \cup \{(p_c, \boxtimes\beta)\}$, where $p_c \in child(p_a) \wedge ancestor(p_c, p)$; if all nodes in $\{p' \mid child(p_a) \wedge \neg ancestor(p', p)\}$ have no child, and the types of all nodes are same as $\boxtimes$, then $t' = (t \setminus \{(p_a p', q') \mid p' \in \mathbb{B}^* \wedge q' = t(p_a p')\}) \cup \{(p_a, \boxtimes\beta)\}$, where $(\alpha = \boxplus \wedge \beta = \circ) \vee (\alpha = \boxplus_d \wedge \beta = \boxplus_c) \vee (\alpha = \triangle \wedge \beta = \circledcirc_c) \vee (\alpha = \odot \wedge \beta = \odot)$;
- $t(p) = q(a)\beta$, where $a \in \mathcal{A}$: $t' = (t \setminus \{(p\theta, \boxtimes\beta), (p, q(a)\beta)\}) \cup \{(p, \boxtimes\beta)\}$.

In all conditions, $s'(m) = s(n)$, and $m = n$.

Extracting the Latent Hierarchical Structure of Web Documents

Michael A. El-Shayeb[1], Samhaa R. El-Beltagy[1], and Ahmed Rafea[2]

[1] Computer Science Department, Faculty of Computers and Information, Cairo University, 5 Tharwat Street, Orman, Giza 12613, Egypt
mikeazmy@yahoo.com, samhaa@computer.org
[2] Computer Science Department, American University in Cairo, Cairo, Egypt
rafea@aucegypt.edu

Abstract. The hierarchical structure of a document plays an important role in understanding the relationships between its contents. However, such a structure is not always explicitly represented in web documents through available html hierarchical tags. Headings however, are usually differentiated from 'normal' text in a document in terms of presentation thus providing an implicit structure discernable by a human reader. As such, an important pre-processing step for applications that need to operate on the hierarchical level is to extract the implicitly represented hierarchal structure. In this paper, an algorithm for heading detection and heading level detection which makes use of various visual presentations is presented. Results of evaluating this algorithm are also reported.

Keywords: heading detection, heading level detection, document structure.

1 Introduction

The hierarchal or logical structure of a document plays an important role in many applications. For example, work presented in [1] exploits the hierarchal structure of a document to carry out anaphora resolution. In [2], the logical structure is used to segment a web document and perform passage retrieval. Other applications that can make use of a document's hierarchal structure include browsers designed for cell phones, PDAs and PCs with non-PC terminals as well as text summarization and data mining applications. However, the hierarchal structure of web documents is not always explicitly represented. Many web designers prefer to use their own styles to represent headings than to use the html heading tags meant to convey a document's logical structure. The work in this paper aims to overcome this limitation by presenting a heading detection algorithm and a heading level detection algorithm through which a document's hierarchical structure can be extracted.

The paper is organized as follows: section 2 presents related work, section 3 details the developed heading detection algorithms; experimental results are reported in section 4 and finally section 5 concludes the paper and presents future research directions.

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 305–313, 2009.

2 Related Work

Extracting the hierarchal structure of a web document does not seem to be an area where much work has been carried out. In [3], an approach is presented for detecting headings in a document to solve the usability problem of browsing web pages designed for PCs with non-PC terminals but the approach does not address the determination of the levels of the detected headings. However, since detecting headings and heading levels can help in segmenting a document to smaller semantically independent units, work carried out in the area of web page segmentation becomes relevant.

A variety of approaches have been suggested for segmenting web documents. Most of these approaches try to make use of the nature of the web documents as a sequence of HTML tag delimited data elements. Many researchers have considered dividing the web document based on the type of the tags encountered in these documents. Examples of the most used tags include the <P> (paragraph), <TABLE> (table), <UL> (list), and <H1>~<H6> (heading) tags. In [4] a web query processing system uses such a tagged based approach to present to the user segments of documents that correspond to his/her query. The Vision-based Page Segmentation (VIPS) algorithm [5] uses spatial and visual cues to segment a web page in an effort to simulate the way a user mentally divides a page when presented to him/her based on his/her visual perception. It utilizes the facts that semantically related contents are often grouped together and that different regions are separated with implicit or explicit visual separators like images, lines, font sizes, blank areas, etc. The assumption of VIPS that semantically related contents are grouped together in a web page is not always true as sometimes semantically related segments may be distributed across a web page. In [6] a pre-trained Naïve Bayes classifier is used to help in the composition and decomposition of the semantically coherent segments resulting from the VIPS algorithm. In [7] an approach whereby template-based HTML web documents are transformed into their corresponding semantic partition trees is presented. This is achieved through the discovery of repeated sequential patterns on the sequence of HTML tags (with their attribute values) in a DOM Tree [8, 9].

Work presented in [10] argues that one of the best ways to view a web page on a small screen device is through thumbnail images that can aid the user to navigate the page by zooming in and out of regions of interest. Web page segmentation is an important part of achieving this effect and the work addresses it from a machine learning perspective as entropy reduction and decision tree learning are used to indicate divides within the page.

3 Heading Detection Algorithms

To infer the hierarchal structure of a document, identification of headings and their relationship with each other must take place. Two phases are employed for carrying out this task:

1. The heading detection phase (identification of headings)
2. The heading level detection phase (identification of relationships between headings)

In the first phase, all document text portions that are recognized as headings are collected and stored along with their features (font size, font weight, color, etc). In the second, a level for each of the identified headings is assigned. Each of these phases is described in details in the following two subsections. The third subsection, explains how results produced by these algorithms can be fine tuned.

3.1 The Heading Detection Phase

The basic idea behind the presented approach is that headings are usually differentiated from other text in the document through the placement of some sort of emphasis on them. So, an important part of the heading detection algorithm is to scan the document whose structure is to be extracted to identify its general features (or what constitutes normal text within it). This step is important in order to understand the significance of various features within various documents as general features can differ from document to document. For example, in a document where most or all of the text is written in bold, having the feature of being bold, will not carry much weight, while it will in another document where the bold feature is rarely used. This step is referred to as the document's 'general feature extraction' step. Generated rules are then applied to text chunks that result from breaking down the document into blocks that share similar features or styles and which are easily detected by a DOM tree [8, 9]. An example of a DOM tree fragment is given in (Fig. 1). Whether or not a chunk/block is considered as a candidate heading is determined through considering whether or not it:

1. has a h1-h6 tag name
2. has font size > general document font size
3. has a font weight > general document font weight
4. has a font color not equal to the general document font color
5. has italic, bold, or underline styles that are not part of the document's general features
6. appears within some text
7. contains block-level elements
8. contains BR or HR tags
9. satisfies a maximum length threshold
10. has words that are all capitalized

The basic algorithm for heading detection is provided in Fig. 2. The algorithm carries out its task through six main steps, two of which have just been described. First, the HTML of a web document is parsed to generate its HTML DOM tree (line 1). Second, different HTML tags and attributes that have the same visual effect are transformed to a standard representation for comparison. For example, the font weight can be represented by the <B> tag, <STRONG> tag or by using "font-weight = 700" attribute. Also the font size can be represented using different measures like pixels and points. So a transformation between the different tags attributes and measures that have the same effect is done by the transform-features function in line 2. Third, the document's general features (such as font size, weight, etc) are identified (line 3). To identify the general features of a document, the system first traverses the HTML DOM tree, and for each node in the DOM tree, it extracts the node's features, and

determines the total number of words that makeup the node's content. Features with the highest number of occurrence (based on the number of words) are chosen by the system to be general features.

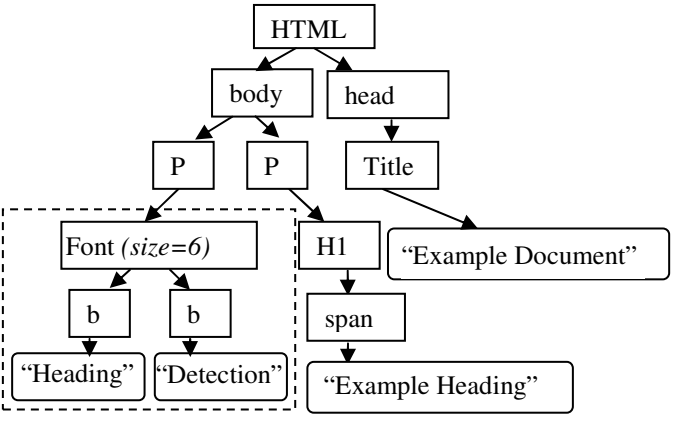


Fig. 1. Example of a DOM tree fragment

Fourth, heading detection rules are built based on the identified general features of the document (line 4).

Fifth, the generated heading detection rules are applied to the DOM tree nodes to generate a list of candidate headings. This task is sometimes complicated by the fact that a single heading can be split across multiple tags. For example in the DOM shown in Fig. 1 the heading "Heading Detection" is represented with the sub tree rooted at a font node (shown within the dashed rectangle in Figure 1).

```
PROCEDURE Detect_Candidate_Headings(Document Doc) {
  DOMTree = parse-html-doc (Doc)                              1
  transform-features(DOMTree)                                 2
  docGenFetures=extract-doc-general-features(DOMTree)         3
  Rules = build-heading-rules(docGenFetures)                  4
  Apply-Rules(DOMTree, Rules)                                 5
}
PROCEDURE Apply-Rules (Tree DOMTree, RuleList rules)
  FOR EACH n ∈ DOMTree DO                                      6
    IF (match(n, Rules)) THEN {                                7
      headingContent = collect-heading-content(n)              8
      IF (validate-heading(headingContent) ) THEN              9
        docHeadings = docHeadings ∪ headingContent            10
      ELSE Apply-Rules(n, Rules)                              11
    }
}
```

Fig. 2. Pseudo-code for the heading detection algorithm

For this reason, rules are first applied to the root nodes of the DOM tree (lines 6-7). If a root node is found to be a candidate heading, then an attempt is made to collect its content from its children nodes through the collect-heading-content function in line 8. If the root node does not qualify as a candidate heading, its children are then traversed in search for headings using the same rules (line 11). Finally, validation rules on the content of extracted candidate headings are applied and a final list of headings is generated. The validation step serves as a final filter for candidate headings. In this step rules such as those that check for a length threshold are applied. Also extracted candidate headings are validated for a manually entered list of text that can not be considered as a heading as part of results fine tuning detailed in section 3.3.

3.2 Heading Level Detection

After the detection of all possible headings in the document, the sequence of features of each heading is then used to detect its level. For example the sequence of features for the "Heading Detection" heading (shown in Fig. 1) is "<font size = 6>, <b>". A hierarchical tree is then built based on the relationship between heading levels. The main root of this tree is the title of the document. The nodes of this tree are the headings of the document. The children of a node are those headings with levels smaller than the heading level of a parent node. The algorithm for heading level detection is given in Fig. 3.

```
PROCEDURE Heading-Level-Detection(HeadingsFeatures
Headings)
  /*first heading is the one with the highest level */
    Headings (0).Level = 1                              1
    CurHeading = 1                                      2
    For each curH ∈ Headings DO {                       3
      curH = Headings (CurHeading)                      4
      prevH = Headings(CurHeading -1)                   5
      Assign-Level(curH, prevH)                         6
      CurHeading = CurHeading +1                        7
    }
}
PROCEDURE Assign-Level (headingFeatures curH, heading-
Features prevH)
  IF curH.isEqualTo( prevH )THEN                        8
    curH.Level = prevH.Level                            9
  ELSEIF curH.isLessThan(prevH) THEN                   10
    curH.Level = prevH.Level + 1                       11
  ELSEIF curH.isGreaterThan(prevH )THEN   {            12
    /* if prevH is not the first heading in the heading
list */
      IF (prevH != 0)     THEN                         13
        Assign-Level(curH, prevH-1)                    14
    /* current heading does not have a parent, so it
will be assigned the highest level */
      ELSE curH.Level = 1                              15
  }
}
```

Fig. 3. Pseudo-code for the heading level detection algorithm

To detect heading levels, we assume that the first heading that occurs after the title of the document has the highest level or a level of 1 (line 1). A comparison is then made between each heading and its preceding heading (lines 4-6). If the features of the current heading are evaluated to be equal to the features of the previous heading, then the level of the current heading is assigned the same value as that of its preceding heading (lines 8-9). When comparing between the features of two headings, weight is given to the features in this order: font-size, font-weight, underlines, and finally italics.

If however, the features of the current heading are less than the features of the previous heading, then the level of the current heading is smaller than the level of the previous heading (lines 10-11). Finally if the features of the current heading are greater than the features of the previous heading, then the parent of the current heading is searched for recursively (lines 13-14). If no parent can be found for this heading, then it is assigned a level of 1 (line 15).

3.3 Results Fine Tuning

Occasionally, web document authors assign headings to text portions just to achieve a certain effect without considering how this will affect the structure of the document. This is particularly common when the author of the document is not a professional web designer. For example, organization or document author names may sometimes appear as headings just because using a heading tag achieves the desired emphasis effect easily. Arbitrary text of certain importance may also sometimes be emphasized by denoting it as a heading. In such cases the results produced by the heading detection algorithm will not be 100% accurate. To detect errors like these, human intervention is required. So, an *option* is provided within a heading detection tool built around the presented algorithms, that enables users from viewing and editing extracted headings and heading levels. Users of the heading detection tool are also provided with the means for entering a list of terms that are often emphasized within documents as headings when they are not in fact that (like organization names). This list can be thought of as a heading stop list. Additionally, more advanced users are given access to dynamically generated heading detection rules, and are allowed to fine tune those.

4 Evaluation

In order to access the overall performance of the developed algorithm, it was applied to a dataset containing a total of 542 headings, of which only 5.3% was represented using html heading tags. Only documents that were unique in the way they represent headings were included in our data set as one of the main goals of carrying out this experiment was to find out how well the system can cope with different presentation styles for headings. As a result, a large number of documents collected from different sites were excluded, as the heading style employed in one document from those sites, was consistent across all documents collected from each. It is important to note that 299 of the headings used in this experiment were written by multiple authors none of which has any understanding of html. These documents were authored using Microsoft word or Front page with the main concern of users being achieving certain effects, which they did in a variety of inconsistent ways. It is also important to note,

that in this experiment, no fine tuning was allowed to take place, which means that the results obtained are those of fully automated heading detection and heading level detection.

The standard information retrieval measures of precision and recall were used to evaluate the success of the algorithm in achieving its task hence forth, the total number of correctly extracted headings will be referred to as true positives (TP), while the total number of incorrectly extracted headings will be referred to as false positives (FP). Given these definitions, in the context of this work:

Precision = TP / TP+FP

Recall = TP/total number of headings in the document set.

The obtained results were also compared to an algorithm that we've devised based on VIPS[5]. The VIPS algorithm, which is a powerful and widely used segmentation algorithm, provides for each segment it extracts, a depth, which means that segments extracted by VIPS are represented in a kind of hierarchy. So, a layer was added on top of VIPS with the objective making this hierarchical representation explicit by assigning a level number to each extracted segment so as to compare the results with our algorithm and to see whether it would be better to go down the path of using VIPS. Basically, in the VIPS based algorithm, if a heading is represented in a segment on its own, then the algorithm is considered to have detected the heading correctly. Heading level detection is then calculated depending on how accurately segments are nested with respect to their parent headings. The results of applying our algorithm as well as the VIPS based one are shown in tables 1 and 2.

Table 1. Results for the heading detection task

	Total Headings	Our Algorithm				An algorithm based on VIPS			
		Recall	Precision	TP	FP	Recall	Precision	TP	FP
Total	542	92.8%	96.4%	503	19	55.5%	96.5%	301	11

Table 2. Results broken down by level for the heading *level* detection task

Heading level	Total	Our Algorithm				An algorithm based on VIPS			
		Recall	Precision	TP	FP	Recall	Precision	TP	FP
Level1	110	87.3%	76.2%	96	30	58.2%	64.7%	64	35
Level2	168	73.2%	69.1%	123	55	32.7%	55%	55	45
Level3	204	75%	81.4%	153	35	11.7%	28.9%	24	59
Level4	45	22.2%	33.3%	10	20	24.4%	42.7%	11	15
Level5	9	0%	0%	0	0	11.1%	25%	1	3
Level6	6	0%	0%	0	0	0%	0%	0	0
Total	542	70.5%	73.2%	382	140	28.6%	49.7%	155	157

In table 1, (TP) or true positives indicate the number of correctly detected headings, while (FP) or false positives indicate the number of wrongly detected headings. In table 2, (TP) indicates the number of headings which were mapped to their levels correctly while (FP) indicates the number of headings which were mapped to their

levels inaccurately. These results seem to indicate that in most cases our proposed algorithm extracted headings accurately. They also show that the proposed algorithm outperforms the algorithm that we've based on VIPS. Cases in which the system falsely detected a heading were mainly due to the misuse of heading tags or of placing emphasis on text in a very similar way to representing a heading. Cases in which headings were not detected, were largely due to headings being represented as ordinary text in the document and cases when heading numbering was used as the only indicator of some text being a heading. The latter case is one which was not handled as focus was placed on presentation features of headings, but is also one that can be easily addressed by a simple improvement in the heading detection rules.

When analyzing the results of the heading level detection algorithm, it was found that undetected as well as wrongly detected headings from the heading detection phase seem to have a major effect in the process of level assignment. This seems logical, as an undetected or wrongly detected intermediate-level headings can propagate incorrect level assignments to detected headings that follow it.

5 Conclusion and Future Work

This paper has presented a novel approach for transforming the implicitly represented hierarchal structure of a web document to an explicitly represented one using algorithms for heading detection and heading level detection. The algorithm presented is simple and straightforward, but the experiments conducted to evaluate this work show that it produces reasonable results. Future work will focus on trying to improve the results of the heading level detection algorithm, and the utilization of text segmentation and similarity techniques along side an ontology to assign headings to segments and infer a document's hierarchy when segment headings are not included explicitly or implicitly in a document.

Acknowledgments. Work presented in this paper has been supported by the Center of Excellence for Data Mining and Computer modeling within the Egyptian Ministry of Communication and Information (MCIT).

References

1. Goecke, D., Witt, A.: Exploiting logical document structure for anaphora resolution. In: Proceedings of LREC 2006 Conference, Genoa, Italy (2006)
2. El-Beltagy, S., Rafea, A., Abdelhamid, Y.: Using Dynamically Acquired Background Knowledge For Information Extraction And Intelligent Search. In: Mohammadian, M. (ed.) Intelligent Agents for Data Mining and Information Retrieval, pp. 196–207. Idea Group Publishing, Hershey (2004)
3. Tatsumi, Y., Asahi, T.: Analyzing web page headings considering various presentations. In: Proceedings of 14th international conference on World Wide Web, Chiba, Japan (2005)
4. Diao, Y., Lu, H., Chen, S., Tian, Z.: Toward Learning Based Web Query Processing. In: Proceedings of International Conference on Very Large Databases, Cairo, Egypt, pp. 317–328 (2000)

5. Cai, D., Yu, S., Wen, J.R., Ma, W.Y.: Extracting content structure for web pages based on visual representation. In: Proceedings of the 5th Asia Pacific Web Conference, Xi'an, China (2003)
6. Mehta, R., Mitra, P., Karnick, H.: Extracting Semantic Structure of Web Documents Using Content and Visual Information. In: Proceedings of the 14th international conference on World Wide Web, Chiba, Japan (2005)
7. Mukherjee, S., Yang, G., Ramakrishnan, I.V.: Automatic annotation of content-rich HTML documents: Structural and semantic analysis. In: Fensel, D., Sycara, K.P., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 533–549. Springer, Heidelberg (2003)
8. HTML 4.01 Specification, http://www.w3.org/TR/REC-html40/
9. http://www.w3.org/DOM/
10. Baluja, S.: Browsing on small screens: recasting web-page segmentation into an efficient machine learning framework. In: Proceedings of the 15th international conference on World Wide Web, Edinburgh, Scotland (2006)

CBR Method for Web Service Composition

Soufiene Lajmi[1], Chirine Ghedira[2], and Khaled Ghedira[1]

[1] SOIE/ University of Tunis, Tunisia
Soufiene.Lajmi@ensi.rnu.tn,
Khaled.Ghedira@isg.rnu.tn
http://www.soie.isg.rnu.tn
[2] Claude Bernard Lyon 1 University, France
Chirine.Ghedira@liris.cnrs.fr
http://www.liris.cnrs.fr

Abstract. The emergence of web services as a new technology supporting the future Web has motivated several researchers to investigate in this field. In addition, the possibility of selecting and integrating Web services, in or between organizations, is very useful for the improvement of their profitabilities. In such a case, we introduce the term of *web service composition*. In this paper, we propose an approach called WeSCo_CBR (Web Service Composition founded on Case Based Reasoning) that aims at enhancing the process of web service composition by using a case-based reasoning technique which originates from artificial intelligence. Furthermore, we regard efficient description and management of information semantics as a major requirement to enable semantic interoperability. We opt to integrate ontology so that we can apply a reasoning to help perform meaningful web service composition.

Keywords: Web Services (WS), Web Service Composition, Case-Based Reasoning (CBR), Ontology.

1 Introduction

With the proliferation of Web services in the last years, distributed architectures have seen a real development. Indeed, not only does WS ensure the interoperability between applications which are supported by heterogeneous systems, but also their composition appears as an important strategy to provide more complex and useful solutions [16].

Composition addresses the situation of a request that cannot be answered by existing and/or elementary WS, whilst it has favored the re-use and the creation of services. However, the simple use of the existing WS standards (such as Simple Object Access Protocol SOAP [1], Web Service Description Language WSDL [15] and Universal Description, Discovery and Integration protocol UDDI [3]) does not ensure a dynamic and efficient composition. In [20], authors have discussed two approaches: industry approach and semantic web approach. For the first one, many research issues, considering process description specifications, have been proposed such as WSFL [5], Xlang [6], BPEL4WS [7].

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 314–326, 2009.

However, none of these suggested specifications actually treat the dynamic business process creation. Indeed, one of the requirements imposed by these specifications is that the process must be pre-defined. For the semantic web solution, several initiatives have proposed languages and frameworks (such as Web Services Modelling Ontology WSMO [14], WSDL-S [17]) which aim at integrating a semantic level to describe web services in order to improve the WS composition process or OWL-S [13] language which has defined *Service class* to model WS with the properties *presents*, *describedBy* and *supports*. OWL-S ontology is divided into three sub-ontologies: the *Service Profile* to describe what the service can do, the *ServiceModel* to describe the control flow and dataflow and the *ServiceGrounding* to specify the details of how a service can be accessed.

Despite these efforts, the WS composition is still a complex task and is beyond the human ability to make the composition plan manually. Yet, semi-automated/fully dynamic web services composition hence presents a real challenge.

A crucial step in the process of developing applications based on WS is the WS discovery. It consists in finding the WS that fulfils better a user request. The main obstacle affecting this step is the huge number of the available WS. A study on the WS discovery mecanisms has been done in [18]. In our work, for the WS discovery, we are only interested in reducing the search space of WS. Thus, the first WS in the space will be selected.

In this paper, we present the WeSCo_CBR approach and demonstrate how CBR techniques are applied for semi-automated WS composition. Our contribution consists of: 1) the definition of an abstract level defined by a set of communities in order to reduce the search space of WS, 2) a method to build an abstract process (through a set of similarity computation procedures according to the CBR techniques) and then a composite WS.

The rest of this paper is organized as follows: section 2 motivates the adoption of the ontology in conjunction with the CBR technique, section 3 deals with the foundations and principles of our approach, section 4 presents an illustrating example and the implementation of the system, section 5 discusses related work, and finally, we present our conclusions and discuss future research in section 6.

2 Background

2.1 Motivating Scenario

Our scenario concerns the medical field. To illustrate the goals of our proposal, we consider a request for a medical diagnosis of the early detection of cardiac ischemia and arrhythmia. This request consists in carrying out a cardiologic diagnosis of a patient, starting from the analysis of his electrocardiogram (ECG). In fact, a patient has a Portable ECG Monitor (PEM), which is used to detect and manage any cardiac event. When the patient feels a chest pain, he turns on the PEM so his ECG is recorded. The PEM starts with a serial analysis of

this record and compares it with the referenced ECG. The PEM service can suspect any cardiac problems and should look for a call center service to send an alert, if needed. The alert triggers a WS whose role is to find a first-aid medical center close to the patient's current location. Processing both the recorded and referenced ECG, the selected medical center identifies the type of alert: severe or minor.

Without the use of communities, the response to this request requires its comparison with the whole content of the repository of the actual WS (even those which do not have any relationship with the medical diagnosis). This explains the difficulty and the high cost of discovery and selection tasks of the actual WS which covers this request.

By community, we mean a concept, described in an ontology, which presents a set of real services having the same functionality. Thus, the fact of using an abstract process (a set of ordered communities) facilitates the discovery and selection tasks by reducing the scope of search. Whenever an abstract process is required, it is sufficient to start, for each community, a search for the WS represented by this community.

2.2 Why Using the Case-Based Reasoning?

Our adoption of the Case-based reasoning is supported by various reasons. First, Case-based reasoning [11,19] is a problem solving paradigm which, in many respects, is fundamentally different from other major AI approaches. Indeed, CBR is the process of solving new problems based on the solutions of similar past problems. In other terms, instead of relying solely on general knowledge of a problem domain, or making associations along generalized relationships between problem descriptors and conclusions, CBR is able to utilize the specific knowledge of previously experienced, concrete problem cases. A new problem is solved by finding a similar past case, and reusing it in the new problem case. Second, CBR is also an approach to incremental, sustained learning, since a new experience is carried out each time a problem has been solved, making it immediately available for future problem solving. Finally, it has been argued that CBR is not only a powerful method for computer reasoning, but also a pervasive behavior in every day human problem solving.

In our study, we propose to apply the CBR method, combined with the use of OWL-S as a language to describe and develop the abstract processes. This type of reasoning consists in finding, in the case base, cases similar to a new user request. In addition, we estimate that the use of ontology is very important for the achievement of a composition platform. Indeed, ontology provides a rich description of the resources which allows to improve the search for the most relevant services and their selections. Moreover, the OWL-S enables the automatic discovery, execution, composition, and interoperability WS [20]. Once the relevant services are selected, the last stage is the construction of a composite web service. In the next section, we present the architecture of WeSCO_CBR.

3 A Proposal for Web Service Composition Based on CBR

In this section, we briefly present the architecture of WeSCo_CBR, next we will proceed with the description of the use of CBR for the semi-automated WS composition.

3.1 WeSCO_CBR

In [4], we have presented the architecture of our proposal which consists of five components: 1) the *Community Discovery Engine* which allows to find the set of communities that best fits a user request, 2) the *Abstract Process Binder* whose role is to build an abstract process made up of a set of communities, 3) the *Web Service Discovery Engine* which allows to determine-for each community of this process-the integrality of the actual semantic WS which can substitute the community, 4) the *Selector* which includes the selection mechanism[1] of one of these discovered semantic WS. This enables to make a comparison between the requested community and the integrality of the semantic WS proposed by the *WS Discovery Engine*, 5) the *Constructor* which allows the transformation of the abstract process built by the *Abstract Process Binder* in an executable process. It is the component which generates an executable OWL-S process that can be executed by the dedicated engine.

3.2 How to Apply CBR in WeSCo_CBR?

In order to simplify the request processing, we need to transform the user request in an easy-to-handle language, understandable by the machine. The reformulation step translates the request in an easy ontological formula. This task is launched by the *Community Discovery Engine*. To do so, we propose to divide the request into three components defined as follows:

- Instances: A request can contain a set of data. These data can be considered as values for attributes of one or more objects. The *Instances* part of the request represents the classes inherited by these objects.
- Variables: A request can contain variables. These variables can be considered as attributes of one or more classes which represent the *Variables* part of a request.
- Communities: It is the set of communities of a request. Indeed, the communities of a request are deduced from variable and instance classes as follows:

Once the *Instances* and *Variables* parts of the request are set, a search for communities is launched by the *Community Discovery Engine*. This engine allows us to obtain all the communities which fit in the request. The community search algorithm shown below is based on the ontological description of the communities for the medical domain classes.

[1] Until now, the first web service among the selected ones will be selected.

Community search algorithm

```
function CCommunityList RequestCommunities(CVariableList,
                                           CInstanceList)
Input:   CVariableList {List of Variable classes}
         CInstanceList {List of Instance classes}
Output:  CCommunityList {List of Communities} Begin
   CCommunityList result=new CCommunityList()
   CVar {Variable class}, CIns {Instance class}
   for CVar in CVariableList do
        result.add(CVar.CommunityList())
   end for
   for CIns in CInstanceList do
        result.add(CIns.CommunityList())
   end for
   return(result)
End
```

For the needs of our study, we have defined a local ontology. It is described in Fig. 1 and presents some concepts and communities and further relationships between them. The *CommunityList* function shown below enables-for each concept-to find communities that have a relation with it.

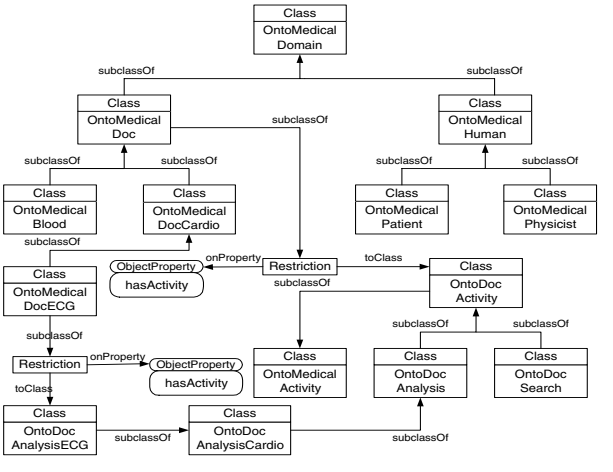


Fig. 1. Ontology illustration of the description of the communities and concepts in the medical domain

CommunityList algorithm

```
function CCommunity[] CommunityList( ) Begin
    OntModel ontology=owlModel.getOntModel();
```

```
    RDFProperty HasCommunity=owlModel.getRDFProperty("hasCommunity");
    String queryString = "SELECT ?domain, ?range\n"
            +"WHERE (?domain  onto:hasCommunity  ?range)\n"
            +"AND ?domain=~/"+CV+"/"
            +"USING onto FOR <http://127.0.0.1/MedicalField.owl#>" ;
    Query query = new Query(queryString) ;
    query.setSource(ontology);
    QueryExecution qe = new QueryEngine(query) ;
    Individual Ind;
    QueryResults results = qe.exec() ;
        for ( Iterator iter = results ; iter.hasNext() ; )
        {
                ResultBinding res = (ResultBinding)iter.next() ;
                Object range= res.get("range") ;
                Object domain =res.get("domain") ;
                Ind=ontology.getIndividual(range.toString());
                Communities.add(Ind.getLocalName());
        }
        results.close() ;
        return(Communities) ;
End
```

As mentioned before, our approach is based on CBR to construct the abstract process made up of the request communities. We believe that an intelligent method can provide better solutions. The CBR technique uses existing cases to provide a process to a user's request. In the first step, the proposed process is the solution of the more similar case to the request. This solution may not use some required services. To do so, we build-in the second step-another request composed of those services. Finally, we obtain two processes. The first process will be adapted with reference to the second. However, the use of CBR requires the identification of a case, which needs to be represented by a model adapted to our problematic. This modelling allows us to describe each component of a case. For the search of similar cases and the selection of the relevant cases, we need the case similarity computation and search procedures.

In the first part of this section, we define the representation of a case. In the second part, we present the methods elaborated to calculate the similarities.

Case representation. According to Kolodner[11], regardless of the applicability domain, a case has always the same components. These components are a t-uple composed of a problem, a solution and possibly an evaluation. Likewise, in WeSCo_CBR [4], a case is composed of the following three elements: 1) *the problem* which is composed of four parts: user's profile, communities, variable classes and instance classes, 2) *the solution* which is the combination of a set of communities, 3) *the evaluation* which is the relevance ratio of the solution. Due to the existence of irrelevant cases which does not fit the user's needs, we propose evaluation criteria of the user to express his satisfaction degree to the suggested process.

After the elaboration of the case modelling, it is necessary to establish the discovery and selection procedures for the most relevant cases. In the following, we deal with the problems of the search and selection of a case for a new request presented by the user.

Similar case search system. For a new request, the re-use process consists in looking for a memorized similar case and, if required, in evaluating and memorizing the new case. Moreover, we need to find, for each request, the most relevant memorized case which can best fit this request. This process is composed of the following stages: 1) *Representation of the problem*: for each new request, we search the most relevant cases. This research is carried out according to the request (problem). To do so, we express the request in the form of a new case in order to be able to compare it with memorized former cases. This stage has been processed in section 3.2, the following stages will be analyzed in this section. 2) *Similarity computation*: the most relevant case is generally given according to its similarity with the new case. With this intention, we define methods for similarity computation in order to guide the research. Accordingly, we propose some methods to calculate the similarity between cases. Similarity computation is done for the components (problem part) of the request. 3) *Procedure of research of the most relevant case*: it uses the methods of similarity computation to propose the most relevant case evaluated by procedures of search and similarity described in this section. 4) *Memorizing*: action left for the user to judge if the new case is interesting enough to memorize. In the same way, we propose to the application user the choice to memorize his new case. The memorized case is followed by an evaluation which will make it possible to refine the following research. In [4], we have presented similarity computation between communities. It is based on two algorithms. The first one allows to calculate the similarity between two communities. The second concerns the case similarity computation in terms of communities. In the next section, we present the similarity computation between variable classes.

Similarity computation between variable classes. The variable classes represent the second dimension for the calculation of similarity between a new case and a memorized case. The similarity between two cases according to their variable classes is expressed by the following formula:

$$Sim_v(NC, MC) = \frac{\sum_{i=1}^{NV} Sim_v(NCV_i, MC)}{\max(NV, MV)} \quad (1)$$

where

- NC is a new case;
- MC is a memorized case;
- NV and MV are, respectively, the number of NC variable classes and those of MC variable classes;
- NCV_i is a variable class of NC.

The similarity between a variable class of a new case and all variable classes of a memorized case is equal to the maximum of the similarities between this class and each variable class of the memorized case. The following formula presents calculation of this similarity.

$$Sim_v(NCV_i, MC) = \max_{j=1..MV}(Sim_v(NCV_i, MCV_j)) \tag{2}$$

where MCV_j is a class of variable of MC.

The similarity between two variable classes is expressed by the following formula:

$$Sim_v(MCV_i, MCV_j) = \begin{cases} 1 & \text{if } MCV_i = MCV_j, \\ x^d & \text{if } MCV_i \text{ is a subclass of } MCV_j, \\ x^{2d} & \text{if } MCV_j \text{ is a subclass of } MCV_i, \\ x^{3d} & \text{else.} \end{cases} \tag{3}$$

d represents the maximum of the distances between NCV_i and MCV_j with their common parent.
$x = (2^*$ the number of common properties between NCV_i and $MCV_j)$ / sum of the properties of NCV_i and MCV_j.

In the case base, we can have a very significant number of memorized cases. However, it can happen that none of these cases answers exactly the request. In this situation, the case selected by the search algorithms is a relevant case but it requires some processing and modifications. In the next section, we present a solution to improve these algorithms.

Case classification. In order to improve the search of the most relevant case, we need to classify all the cases. For this, we opt to use an Ascendant Classification Algorithm shown below. This algorithm starts its execution from a single class composed of the whole of cases and then divides this class into two classes. The same processing is repeated for the new obtained classes until having a desired number of classes. This algorithm maximizes the inter-class distance and minimizes the intra-class distance. We have defined the distance between a case c and a class Cl as follow:

$$D(c, cl) = \sum_{c_i \in cl} d(c, c_i)$$

where $d(c, c_i)$ represents the distance between tow cases c and c_i and depends on the distances between different components of these two cases.

Classification algorithm

```
Procedure  Classification(StartClass, ClassNumber)
Input:   StartClass {StartClass contains the whole of cases}
         ClassNumber {Number of classes}
Output:  ClassSet {Set of Classes} Begin
```

```
   Cn Integer=1
   while Cn <> ClassNumber do
        {Create the Cn+1 class}
        Select the case c where D(c,cl)= max(D(c',cl') and c' in cl'
                                         c'
                                         and c in cl
        Class(Cn+1)<-- c  {insert c into the new class number Cn+1}
        while there are a case c where D(c,cl) - min(D(c,cl')>0 and
                                               cl'
          D(c,cl)- min(D(c,cl')>0 = max(D(e,cle) - min(D(e,cle)))
                   cl'              e             cle
          {where c in the class cl and the case e in the class cle
           and c is more near to cl' than its class cl }  Do
             cl'<-- c
        end while
        cn <-- cn+1
   end while
End
```

After the execution of the classification algorithm, we need to have a solution to find the most similar case to the new case from the case base. Accordingly, it is necessary to find the class where we can find this case. For this purpose, we should organize the case base in the best way. We propose to use the Galois lattice of a binary relation. Given tow sets E and E', and a binary relation R between these two sets, $R \subseteq E \times E'$, we can represent by a lattice the gathering of elements of E and E' according to the relation R. This structure is named Galois lattice. Each element of the lattice is a pair, noted (X, X'), composed of a set $X \in P(E)$ and a set $X' \in P(E')$. P(A) represents the powerset of A. Each pair must be a complete pair as defined in the following. A pair (X, X') from P(E) x P(E') is complete with respect to R if and only if the two following properties are satisfied:

1. X' = f(X) where f(X) = $\{x' \in E' \text{ — } \forall x \in X, \text{xRx'}\}$;
2. X = f'(X') where f'(X') = $\{x \in E \text{ — } \forall x' \in X', \text{xRx'}\}$.

The following references give a more details and formal aspects of the Galois lattice [22,23]. In the next section, we present an illustrating example and an overview of the implementation details.

4 Illustrating Example and Implementation

In this section, we illustrate WeSCo_CBR by using an example. An overview of the implementation details, then, is presented. In the following, we consider a set of cases $\{ci\}_{i\in[1..9]}$. The execution of the classification algorithm allows us to get the class hierarchy composed of five classes {A, B, C, D, E}. The distances between these cases and the class hierarchy are presented in figure 2.
A=$\{ci\}_{i\in[1..9]}$; B=$\{c_1, c_2, c_7, c_8, c_9\}$; C=$\{c_3, c_4, c_5, c_6\}$; D=$\{c_1, c_2\}$; E=$\{c_7, c_8, c_9\}$

	C1	C2	C3	C4	C5	C6	C7	C8	C9
C1	0	1	5	6	4	5	4	4,5	5
C2	1	0	6	7	5	6	3	4	4,25
C3	5	6	0	1	2	2	4	3,8	7
C4	6	7	1	0	1	1	5	3,5	6,75
C5	4	5	2	1	0	2	4	5,5	6
C6	5	6	2	1	2	0	5	6	6,5
C7	4	3	4	5	4	5	0	1	1,5
C8	4,5	4	3,8	3,5	5,5	6	1	0	2
C9	5	4,25	7	6,8	6	6,5	1,5	2	0

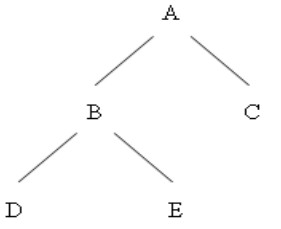


Fig. 2. Distance table and class hierarchy

	C1	C2	C3	C4	C5	C6	C7	C8	C9
B	1	1	0	0	0	0	1	1	1
C	0	0	1	1	1	1	0	0	0
D	1	1	0	0	0	0	0	0	0
E	0	0	0	0	0	0	1	1	1

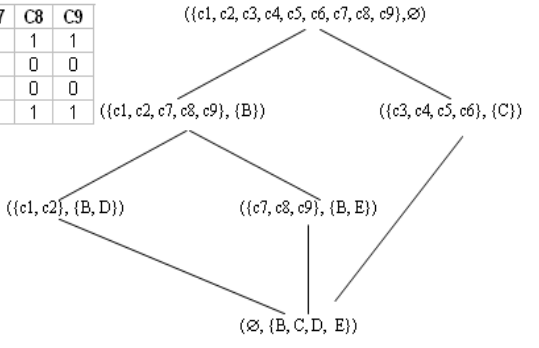


Fig. 3. The relation matrix representation and its Galois lattice

After the classification of all the cases, we need to built the Galois lattice from a relation matrix representation. For this, we consider the relation R defined as follow:

$$cR\mathcal{C} \Longrightarrow c \in \mathcal{C}$$

where c is a case and $\mathcal{C}$ is a class in the class hierarchy.

The binary relation R is expressed by a matrix. Figure 3 shows this matrix and its corresponding Galois lattice. Accordingly, several building Galois lattice algorithms were proposed. We have chosen the "Incremental Structuring of Knowledge Bases" which is an algorithm presented in [21].

Finally, the implementation of all components was realized by jdk 1.5.0 using JBuilder 7.0. as a development environment. We also used existing semantic web tools, like OWL, OWL-S, Protege 3.0 and Jena 2.1 to provide the basic representation and reasoning technology. Communities, Variable classes and Instance Classes are described using OWL whereas OWL-S was used to create semantic WS. The cases were saved in a MySQL database. The *Constructor* uses the OWL-S API to create a composite WS from an abstract process provided by the *Abstract Process Binder*. To reason on the ontology, we have chosen to use Jena 2.1 which is a Java API for semantic web applications. This API deals with searching and reasoning on ontology like the ones we have defined. To demonstrate the feasibility of our proposal, we have developed a graphical user interface that allows to enter a request. The user can launch the reformulation request process. An OWL file is generated to describe the different components of the reformulated request. This

OWL file is then used by the *Abstract Process Binder* to generate the solution of the user request.

We can show the solution proposed by our prototype. It can be modified manually by the user and adapted by creating another request for a new sub-problem.

5 Related Work

Recently, several approaches to applying AI techniques to the web service composition problem have been published. Other approches are based on the workflow. EFlow[8] uses a static workflow generation method. A composite service is modelled by a graph that defines the order of execution. In Meteors [9], the authors have proposed an approach which consists in adding the semantic to the current standards such as UDDI, WSDL and BPEL. However, those two initiatives require a predefined workflow, which can be a major disadvantage. Some other work is based on the technique of artificial intelligence planning. SWORD [10] presents one of these works. However, SWORD does not use the emerging service description standards. [15] has presented an approach that supports running views over the specification of a composite web service. This approach is based on context and constraints to generate a derived state chart diagram[2] from another state chart (initial or derived). In fact, if the constraints on an incoming transition of a service chart diagram is not satisfied in a certain context, then this service chart diagram will be excluded from the derived state chart diagram. However, a derived specification does not accept extra services through their service chart diagrams. In our proposal, we suggest to use an approach based primarily on the Case Based Reasoning (CBR). This approach has been dealt with in paragraph 3.2. Accordingly, we can integrate existing cases to generate a composite web service which responds to the user request. However, the selection of the first discovered web service can provide an unsuitable solution. So, the selection mechanism must be improved.

6 Conclusion and Future Work

Recently, several technologies and languages interfering in various stages of WS life cycle have been developed. However, those single technologies do not allow an effective and dynamic composition of WS. For this purpose, WeSCo_CBR combines the CBR technique and the semantic description of Web services. It is characterized by the two advantages. On the one hand, the use of ontology and description semantics of the communities and the services allowing the best reasoning in the various steps of the composition. On the other hand, it allows a dynamic composition of WS starting from a user's request. Our future works will focus primarily on the distribution of WeSCo_CBR using a multi-agent system. Indeed, we estimate that the distribution of the repository of communities, as

[2] A state chart diagram is used as a means for modelling and specifying the component web services of a composite service.

well as the repository of the WS is very important. Therefore, we propose to use intelligent agents to decentralize the WS composition process.

References

1. Moreau, J.J.: Introduction to soap awalkthrough of core technical concepts. In: XML EUROPE (2004)
2. Mantek, F.: What's new in wsdl. In: Wrox Conferences (2002)
3. Chauvet, J.M.: Services web avec soap, wsdl, uddi, ebxml, vol. 99, p. 524. Eyrolles, Paris (2002)
4. Lajmi, S., Ghedira, C., Ghedira, K., Benslimane, D.: Wesco_cbr: How to compose web services via case based reasoning. In: The IEEE International Symposium on Service-Oriented Applications, Integration and Collaboration held with the IEEE International Conference on e-Business Engineering (ICEBE 2006), Shanghai, China (October 2006)
5. Leymann, F.: Web services flow language (wsfl 1.0) (May 2001)
6. Levy, D.: Coordination of web services: langages de description et plate-formes d'exécution (Septembre (2002)
7. Juric, M., Sarang, P., Mathew, B.: Business process execution language for web services. p. 270 (October 2004)
8. Casati, F., Ilnicki, S., Jin, L., Krishnamoorthy, V., Shan, M.-C.: Adaptive and dynamic service composition in eFlow. In: Wangler, B., Bergman, L.D. (eds.) CAiSE 2000. LNCS, vol. 1789, p. 13. Springer, Heidelberg (2000)
9. Aggarwal, R., Verma, K., Sheth, A., Miller, J., Milnor, W.: Constraint driven web service composition in meteor-s. In: 2004 IEEE International Conference on Services Computing (submitted to, 2004)
10. Ponnekanti, S.R., Fox, A.: Sword: A developer toolkit for web service composition. In: Proceedings of the 11th World Wide Web Conference, Honolulu, HI, USA (2002)
11. Kolodner, J.L.: Case-based reasoning. Morgan Kaufmann, San Mateo (1993)
12. Limthanmaphon, B., Zhang, Y.: Web service composition with case-based reasoning. In: Proceedings of The 14th Australasian Database Conference (February 2003)
13. Burstein, M., Ankolenkar, A., Paolucci, M., Srinivasan, N.: Owl-s: Semantic markup for web services (2003)
14. Arroyo, S., Stollberg, M.: WSMO primer. SMO Delivrable D3.1, DERI Working Draft, Technical reportl (2004)
15. Benslimane, D., Maamar, Z., Ghedira, C.: A view based approach for tracking composite web services. In: ECOWS, Växjö, Sweden. IEEE Computer Society, Los Alamitos (2005)
16. Ghedira, C., Maamar, Z., Benslimane, D.: On composing web services for coalition operations - concepts and operations. International Journal Information & Security. Special issue on Architectures for Coalition Operations 16, 79–92 (2005)
17. Miller, J., Verma, K., Shelth, A., Aggarwal, R., Sivashanmugan, K.: WSDL-S: Adding semantics to wsdlwhite paper. Technical report, Large Scale Distributed Information Systems (2004)
18. Garofalakis, J., Panagis, Y., Sakkopoulos, E., Tsakalidis, A.: Web Service Discovery Mechanisms: Looking for a Needle in a Haystack? In: International Workshop on Web Engineering (2004)

19. Leake, B.: CBR in Context: The Present and Future. In: Leake, D. (ed.) Case-Based Reasoning: Experiences, Lessons, and Future Directions. AAAI Press/MIT Press (1996)
20. Srivastava, B., Koehler, J.: Web Service Composition Current Solutions and Open Problems. In: Workshop on Planning for Web Services ICAPS (2003)
21. Godin, R., Mineau, G., Missaoui, R.: Incremental structuring of knowledge bases. In: Ellis, G., Levinson, R.A., Fall, A., Dahl, V. (eds.) Proceedings of the 1st International Symposium on Knowledge Retrieval, Use, and Storage for Efficiency (KRUSE 1995), Santa Cruz (CA), USA, pp. 179–193. Department of Computer Science, University of California at Santa Cruz (1995)
22. Barbut, M., Monjardet, B.: Ordre et Classification. Algébre et Combinatoire, Tome II. Hachette (1970)
23. Davey, B.A., Priestley, H.A.: Introduction to Lattices and Order. Cambridge University Press, Cambridge (1992)

Binding Structural Properties to Node and Path Constraints in XML Path Retrieval

Gildas Ménier, Pierre-Francois Marteau, and Nicolas Bonnel

Université de Bretagne Sud,
56000 Vannes, France
{gildas.menier,pierre-francois.marteau,
nicolas.bonnel}@univ-ubs.fr

Abstract. We describe a path approximate search process based on an extended editing distance designed to manage 'don't care characters' (*) with variable length (**i~j*) in a path matching scheme extending XPath. The structural path is bounded to conditional properties using variables whose values are retrieved thanks to a backtracking processed on the editing distance matrix. This system provides a dedicated iterator for a XML query and processing scripting language that features large XML document collection management, joint operations and extraction features.

Keywords: XML, data management, query and processing language, approximate search, editing distance, XPath, XQuery.

1 Introduction

Many of the XML systems use path processing instead of tree matching, both to lower the processing cost and because the fragment [12] approach is often enough satisfying for the user. Since the user may not have enough knowledge about the structural properties of the collection of semi structured documents, it seems acceptable to provide an approximate search scheme that does not strictly rely on exact path matching, but also on partial path matching. Many extensions have been made to XQuery [31] or to XPath [30] (TeXQuery [3, 6], XQuery/IR [7], ELIXIR [8], XIRQL [13], XXL [24], XRank[14,23] and XIRCUS [19] for instance) to meet partial matching. FleXPath [2] is one of the most interesting extensions proposed to the field. XML-QL [29] is a proposed W3C standard for queries that involve both an exact matching and an approximate search extension proposed by INRIA. One of the most delicate things to design is the expressiveness of the request language: it should help the user to define precisely in his request, where the approximation is acceptable and where it is not : the user can then express requests that vary from 'very' approximate to 'strict' matching along its document base discovery. In this view, a query language should be able to morph from an exploring tool to a strict search engine for content and structure [4].

Many string processing have been studied in the field of path matching. Many of them, dealing with approximate string matching have been used to provide approximate path search. In the *Levenshtein* [16] editing distance for instance, a global cost

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 327–338, 2009.

can be used to compute a distance. This has been studied for approximate path matching in XML search [18] and leads to encouraging results [21] – even if the computation of joined XML match and attribute condition remains delicate to express as editing costs. In this scheme, it is uneasy to express which part of the path should strictly match and which part doesn't have to. The use of suffix trees [28] for efficient indexing or management of '*' or 'don't care characters' is another way to manage imprecise path matching [9]. The expression of unknown regions lays then in the use of these '*', the other parts having to match strictly. This kind of algorithm has been studied extensively [11].

In the *part 2 'problem'*, we will state the problem of path search in relation to elements and attributes properties. Then, in *part 3 'Q'Path$_{structural}$ distance computation*', we describe a matching operator able to express local approximation by the mean of both 'variable length don't care' and editing distance operators. We present the algorithm used beneath this process. In *part 4 'Retrieval of structural bonds: $S^{B'}$,* we show how the binding to DOM Tree nodes property constraints is managed. We explain an indexing process to deal with this search and introduce then (in *part 5 'Processing System'*) the Alpha system which features this dedicated distance and t-uple assignment. We introduce search and processing results and then conclude.

2 Problem

In the common query languages, most of the semi structured requests involve both a path and properties on the elements in the path. These properties can be seen as constraints or conditions on attributes or words that may be found in the node or path context. This is the case for instance for XPath [30], or NEXI language ([25]). Most of the time, the search involves an inverted list which gives access to the list of path that satisfies the conditions. This approach achieves a high speed retrieval but cannot always apply when only structural properties are formulated (an inverted list entry is not always provided).

In this work, we focus on the search of a node structure first, and then, on the fulfillment of conditions (if any) on the node unification: this ordering may not be always optimal in specific cases but it ensures that any conditions can be expressed and generalized.

Each XML document 'd' can be depicted as a tree T_d in which each node $N(T_d)$ is represented by a tag – or a XML element.
Let $Path(T_d)$ be a path of T_d, also a sequence of Nodes $N_i(T_d)$.
Let $S_{path}(T_d)$ be the set of all possible path in this tree T_d.
Let $SC_{path}(C)$ be the union of all $S_{path}(T_k)$ for T_k belonging to a collection C of XML documents.

Let $S_{prop\_node}(N)$ be the set of all computable properties Prop_node(N) on a node N. This set contains (for instance) properties on attributes (an attribute exists or not), on attribute values, on text (if the node has some textual data of course).

As stated before, a path $Path(T_d)$ is a 'word' built on the set of 'letters' $D(T_d)$ of the elements nodes (empty or not) of T_d. Thus, any path can be written as a sequence $N_{root}/N_1/\ldots/N_i/\ldots N_{end}$. $Ni \in D(T_d)$. In the same way as for a node, a path $Path(T_d)$ has

some properties. For instance, its length can be computed. As the path belongs to a specified document, this document's name or path can be found etc..
Let $S_{prop\_path}(P)$ be the set of all computable properties Prop_path (P) for path P.
Let S_{QPath} be the description of a subset of Path in (C): A path P belongs to S_{QPath} if , and only if, it features the conditions described in QPath. QPath is a sequence of conditions Cond() both on node - $Cond_{node}(N)$ - and on path/ subpath - $Cond_{path}(N_i/N_{i+1}/\ldots N_j)$ -.
QPath is practically speaking a pattern with conditions on the related parts of the path. For instance,
$QPath_1$ = [node with name="article"]/ [a path with 3 undefined elements] / [node with attribute date]
is a sequence of three conditions, the first and the last involve properties on nodes, whereas the second involves a property on a sub path. For the following document (and the following nodes):

```
<article author="bob">
      <document>
            <chapter num="3">
                  <text>
                        <paragraph date="3 sept 2000">
                        </paragraph>
                  </text>
            </chapter>
      </document
</article>
```

The following path would belong to S_{QPath1} :

/article/document/chapter/text/paragraph

The subpath */document/chapter/text* has a length of 3. <*article*> has a property 'name' (its own name) with a value of 'article'. <*paragraph*> has an attribute named 'date'.

In the most obvious cases, the set of properties for a node is strongly related to the node's name. If there is an imprecise match for an element's name, there will probably be a bad matching for its attributes names, and of course, values: but even if there is a perfect match for the element's name, one cannot systematically assume that the semantic of the element is the same. Nevertheless, we made the assumption that the most critical approximation in such a path remains the node's name. Thus, we focus on the *name*'s property first.

As for the sub path, since a sequence of nodes 'inherits' a sequence of nodes properties, we focus on a subset of path specific features, and especially on the length feature.

For a given QPath, we also want to find an approximate set of path-solutions that match 'as best as possible' the conditions Cond, with eventually a rejection limit given by the user.

In the scope of these assumptions and restricting the QPath to the features we want to focus on, we introduce the following sub path condition **i~j* that means 'any

sequence of undefined elements that have at least *i* elements and at most *j* elements'. We also define its extension: * or *0~∞ (respectively for a path with zero elements and for any number of undefined elements), ? or *1~1 (one undefined element, or an undefined path of length 1). (Compared to XPath for instance, the // axis would be expressed as *0~∞ . The *1~1 should match the * from XPath)

Thus, $QPath_1$ can be rewritten as $Q'Path_1$:

*$Q'Path_1$ = [article] [*3~3] [node with attribute 'date']*
Or
*$Q'Path_1$ = [article] [*3~3] [node := ? (node has an attribute 'date')]*
Or
*$Q'Path_1$ = [article] [*3~3] [thisnode := ?]*
with the condition (thisnode has an attribute 'date')

Under these assumptions, Q'Path becomes a sequence of structural properties in {element's name, *i~j} with a set of local conditions (in the example, see after *with the condition*). The structural properties are bound to the condition, using references – here 'thisnode' is a reference bound to the node found for the match of '?'.
So we practically split Q'Path into $Q'Path_{structural}(S^B).Q'Path_{conditions}(S^B)$ where S^B is the set of bounding references. In this example S^B = {bound (thisnode,?)}

In the example provided above,
Q'Path =
$Q'Path_{structural}(\{bound(thisnode,?)\})$. $Q'Path_{conditions}(\{bound(thisnode,?)\})$
with
$Q'Path_{structural}(\{bound(thisnode,?)\})$ = *[article] [*3~3] [?]*
and
$Q'Path_{conditions}(\{bound(thisnode,?)\})$ = *thisnode has an attribute 'date'*

To evaluate a match between a path candidate - *Pcandidate* - and Q'Path, we intend to find at first an approximate matching of the structural properties $Q'Path_{structural}$, retrieve the structural bonds S^B (ie solve the unification problem of the bound name with the value found) and then deal with the conditions $Q'Path_{conditions}$ on S^B.
In the following part, we'll introduce the algorithm used to perform the match and distance evaluation to find the best structural match. This algorithm is based on an editing distance computation, designed to take the **i~j* into account.

3 $Q'Path_{structural}$ Distance Computation

3.1 Editing Distance with *i~j

The editing distance has been extensively studied [11] and many algorithms have been developed to compute a distance as the sum of elemental editing [16] cost (mainly insert, delete or replace) to get from a string to another string [15]. This involves mainly the computation of a complete or partial cost matrix, thus leading to a

complexity of o(nm) with n and m the length of the two strings. Unlike more traditional approaches, the editing distance takes into account the insert operator that may be discarded in Hamming distances for instance.

The use of * has been also extensively studied and many solutions [9,19] have been developed and lead to high performance algorithms. These solutions deal mainly with exact match of part of strings (separated by *) and don't fit well to our problem. We intend to use an implementation of an editing distance merged with variable length * management that takes into account a 'don't care path' extension (**i~j*). We use then this *-editing distance to solve the $Q'Path_{structural}$ problem stated above.

Few works [17, 33, 1] have addressed the merging of the two problems. In our case, the approximation has also to take into account a possible error on the *i~j* extends provided by the user. This error, as insertion or deletion (but of course no substitution) should be expressed as extra or missing symbols to fit into the *i~j* interval.

The cost computation in the (standard) editing distance matrix involves the evaluation of a minimal cost path at each position (o, p). For standard editing distance, the cost is computed as follows (1):

$$Dist([o,p]) = \min([o-1,p]+1,\ \ [o,p-1]+1,\ \ [o-1,p-1]+sub(o,p)) \quad (1)$$

where sub(o,p) is 0 if the current string character (position o) matches the pattern character (at position p), 1 otherwise. In standard editing distance, the pattern symbol is a letter. In our case, it may be a **i~j*. In this last case, the computation of the Dist[o,p] involves the computation of a minimum cost for the set of hypothesis unfolding the set of j-i possible match. Let call '?' the 'any' symbol that can stand for *1~1. This special symbol has no substitution cost for any symbol (character in the case of a string) of the string to match. So the computation of the matrix related to *i~j involves the computation of the collection of cost sub matrixes with *i* occurrences of ?, *(i+1)*.?, *(i+2)*.?...*(j-1)*.?, *j*.?. Then these sub matrixes are merged to provide the path with a minimal cost possible.

A starting inserting cost very high for the pattern **i~j* has to be set to prevent the algorithm to discard the * constrain inserting '**i~j*' in the string to analyze.

```
D[][] is defined as an array of integers;
Q'Path(structural) is an array of symbols that contains element's names
or *i~j symbols
P(candidate) is an array of symbols that contains only element's names:
it codes a path that could match the pattern Q'Path(structural).

n = length of P(candidate) ;  m = length of  Q'Path(structural);

ip=1; while((ip<=m) and (Q'Path(structural) [ip-1]<>'*')) ip++;
// finds the position m1 of the first * pattern side
if (ip<m) m1 = ip; else m1 = ∞; // m1 is this start, ∞ means no '*'

for(jp=0; jp<=n; j++) D[0][jp] = jp; // initialize the cost matrix
// for this column, this is the same as for a standard cost matrix

jp = MIN((m1-1),m); for(ip=0; ip<jp; ip++) D[ip][0] = ip;
// This computes the cost for the first row : it takes into account
// the don't care characters

for(ip=jp; ip<=m; ip++) D[i][0] = ∞; // then ∞ to prevent false insert,
// ie : we don't want the algorithm to discard the * characters, so we
// introduce a high value (cost) here
```

```
for(ip=1; (ip<=m); ip++) { // cost matrix computation main loop

    if (Q'Path(structural)[ip-1]=='*i~j') {// a '*i~j' in the struct.
pattern

           for(jp=0; jp<=n; j++) D[ip][jp] = ∞;
           // prevents the discarding of '*' – using a high cost.
           // the following inner double loop computes the result of the
merging of all the cost sub matrix computed for each hypothesis of un-
known symbols
           for(jp=0; jp<=n; jp++) {
              for(h=(jp+i); h<=(jp+j); h++) {

           // merges the unfolded computations for hypothesis of
           // i ?, or i+1.? or i+2.? etc… to j.?
           // only keeps the minimal path cost :

              if ((h<= n) and (D[ip-1][jp]<D[ip][h]))
                              D[ip][h] = D[ip-1][jp];
              }
            }

    } else { // a legal element in the structural pattern (<> *)
             // Here we have the common editing distance algorithm
            for(jp=1; jp<=n; jp++) {
              D[ip][jp] =    MIN(   D[ip-1][jp]+ 1 ,
                                    D[ip][jp-1]+ 1 ,
                                    D[ip-1][jp-1]+
               SUBST(Q'Path(structural) [ip-1], P(candidate) [jp-1]) );
// SUBST is the cost (0 or 1) of the substitution of an element of the
candidate by an element of the pattern
           }
      }
}
return D[m][n]; // this value is the minimal path cost computed
```

For a j requested at ∞, we used the value max(n,m) – with n is the length of $P_{candidate}$ and m is the length of $Q'Path_{structural}$. Given a $Q'Path_{structural}$ and a path candidate $P_{candidate}$, this pseudo-editing distance computation scheme provides a minimal cost or number of transformation to get from the path $P_{candidate}$ to the $Q'Path_{structural}$ taking into account the **i~j* symbols. Note that this isn't really a distance anymore since it lacks symmetrical feature. The user may provide a maximum distance allowed: in this case, the computation is aborted as soon as the minimum distance computed reaches this limit. In this case, the pseudo-distance is ∞ ($P_{candidate}$ is rejected).

When no '*' is present in $Q'Path_{structural}$, then the above algorithm reduces to the standard editing distance algorithm of o(nm) complexity. In the worst case, when $Q'Path_{structural}$ is only made of 'don't care characters' *0~∞ (which is not realistic since it can reduce to a single **i~j*) then the complexity is of o(m.n.max(n,m)). So if the length of Path is of order L, the complexity is of $o(L^3)$.

Using a suffix tree to sort the 'don't care column' and merge the 'unfolded *', it is possible to simplify the inner double loop; reducing this complexity to $o(L^2\log(L))$. In the worst case, the mean length of path is 8-10, so in the very worst cases, L^3 is between 512 and 1000 and $L^2.\log(L)$ is between 60 and 100, that is acceptable since the most important time cost for the processing of nodes remains the storage input/output bottleneck.

3.2 Indexing and Search Process

In order to reduce the processing if this editing distance, we have to put aside the path candidates that won't obviously match the search. In this view, we use a path signature to prune the path candidates search. What we call a path signature is actually a string made of the sequence of the first letter of each element in the path. For instance, the path /**a**rticle/**d**ocument/**c**hapter/**p**art has the signature *Signature*(/**a**rticle/**d**ocument/**c**hapter/**p**art)='adcp'. We proceed similarly for the $Q'Path_{structural}$ that provides a kind of pattern signature: For instance, *[article] [*2~2] [?]* has the pattern signature 'a*2~2?' – where '*2~2' is considered as one signature's symbol – just as 'a'.

We use a similar editing-distance to compute a pseudo distance between 'adcp' and 'a*2~2?'. Of course, one cannot assume that the result would be the same as for the full path computation even if the * are legally taken into account (as a character stands for an element). In fact, if there is a difference between a first letter and its 'first-letter' counterpart in pattern, then, one can be sure that the 'extended' element's name are different (as their first letters are not the same). This obviously rejects the element's names that differ only by the first letter.

Hence: $P_{distance}(Q'Path_{structural}, P_{candidate}) \geq P_{distance}(Signature(Q'Path_{structural}), signature(P_{candidate}))$.

$SC_{path}(C)$ (the set of all possible paths in the base) is clustered between classes represented by the different signatures. When a search is performed, a pseudo distance is computed between each of the class signatures and the signature of $Q'Path_{structural}$. Given a maximum acceptable pseudo distance $MaxP_{distance}$, it is possible to discard all the $P_{candidate}$ that belong to class which signature has been rejected.

4 Retrieval of Structural Bonds: S^B

After a pseudo-distance between a $Q'Path_{structural}$ and a $P_{candidate}$ has been computed, the structural/conditional bond references must be retrieved to evaluate the conditions $Q'Path_{conditions}(S^B)$. For instance, in *[article] [*3~3] [thisnode := ?]* the element matching '?' should be retrieved and referenced by 'thisnode'. 'thisnode' is then used as an alias/variable to evaluate conditions (if expressed) on the '?' element.

The matrix used by the *-pseudo-editing distance computation is stored and used to backtrack the possible matching paths. The algorithm starts by the last value computed and back-searches each of the possible paths leading from the first cell (top-left) to the bottom-right cell. This algorithm takes into account the (i,j) values for each of the 'don't care character *'. Each diagonal move (x,y)->(x+1,y+1) relates to a substitution between the member x of the pattern (an element or a **i~j*) and an element of the path candidate

If the member 'x' of the pattern has a bond reference ('b:' in …/b:x/…), then this reference is linked to the substitution retrieved – or to the sequence of substitutions for a path. For instance: /article/document/*/chapter/?/ has no bonds. Thus, no retrieval has to be performed since no condition $Q'Path_{conditions}$ can be expressed.

/*first*:article/document/*list*:*/chapter/*element*:?/ has 3 bonds {bound(*first,<article>)*, bound(*list,**), bound(*element,?)*}.

Each matching path can be translated into a path of node references (that points to an entry in the index base). For a given acceptable path $P_{candidate}$, it is hence possible to find node reference values for *first*, *list* and *element*. In this above example, *first* and *element* are related to nodes, whereas *list* is a reference of a list of node references (eventually empty).

This backtracking process generates all the possible sets of values for {*first, list, element*}: for instance this may be {(first->node324, list-> node345/node654/node4456, element->node45), (first->node32, list->empty, element->node456)} is an example of set of substitutions computed for a given cost matrix. Because the conditions have to be evaluated on the nodes properties pointed by these bonds, no errors are allowed on elements that carry a bond. This is a way to specify, in the Q'$Path_{structural}$, whether an error should be acceptable or not (it is legal to use a bond without any further condition : this rejects the $P_{candidate}$ with a cost substitution on this element). This processing is only performed on the final accepted $P_{candidate}$. The retrieving of the tuple of variables is also of a complexity under $o(L_{path}.\log(L_{path}))$, with L_{path} the length of a path.

The final evaluation of a path is performed on the resolved bonds, using the properties S_{prop_node}(Node) for the nodes or/and S_{prop_path}(Path) for the bonds relating to list of nodes (such as 'list:*2~4').

5 Processing System

We decided to design our own system, similar to XQuery (and XPath), but able to manage a large collection of document (see also [32]), both for the search, navigation, extracting and processing. The system (called *Alpha*) manages a cluster of computers for the indexing and the retrieving of a large collection of documents (the system has been tested with 10 Gb XML Documents). The indexing scheme uses several QDBM databases on disk [26] as well as a client/server built in Visual Prolog [22,27].

The scripting language – *alphaScript* - provides many access levels starting with low-level Document Object Model access. The scheme described in this paper is used to provide a node level location access inside an iterator designed for processing and extraction.

The processing can involve document management as well as indexing, XML document generation from other databases or from internet.

For instance, `_system.processdir("xml","*.xml","memidx");` triggers the indexing of the collection of XML files in the directory *'xml'* to build an index database called 'memidx'.

The following example features a script that generates a new XML document using an already indexed database 'memidx' :

```
_io.setExtractFile("travel92.xml");
_io.extract("<?xml version="1.0" encoding="UTF-8"?>") ;
_io.extract("<travels>") ;

foreach  /a:article/*2~3/b:chapter/txt/part/abstract/c:?  max 1
   in "memidx"
   where ( a.date>"1992")
            and (b.num==5)
            and (c.containsWord("travel"))
```

```
{
  _io.extract("<desc document='", a.sourceFile,"'>", c.text, "</desc>");
}
_io.extract("</travels>") ;
```

This script produces a new XML document with information extracted from the index database *memidx*. It is a feature already offered by (for instance) XML-QL, or XSLT. It is possible to produce any kind of text document (HTML, SVG or XML) and have the script produce another index database from the collection of documents made. This way, starting for instance with a collection of document on movies with all details on actors, soundtrack etc, it is possible to generate or extract a new XML collection of document dedicated to the actors only and build a more specific index database for faster search & processing.

Note that each variable is a reference of the corresponding DOM node: it is hence possible to gain access to the node properties. For instance, the variable 'c' is set to the matching element for '?: *c.name* gives also the name of the unknown element. Unlike XQuery (but like in XML-QL [29]) it is possible to express queries based on the name (or attributes) of elements and not only on their contents.

This XPath-like *foreach* is not limited to non branching path management: not only it is possible to use the DOM-like access on the nodes (scanning thru any connected node), but *alphaScript* supports also joint-like operations on nodes as well (or based on any node or path properties):

```
foreach /*/root1:?/aaa in "idxmem" {
   foreach /*/root2:?/bbb in "idxmem" where (root1==root2) {...}}
```

describes two elements, aaa and bbb that have the same ancestor.

```
foreach /path:*/aaa in "idxmem" {
    foreach /*/root2:?/*/bbb in "idxmem" where (path.contains(root2))
{…}}
```

constrains bbb to have one ancestor that belongs to the list of ancestors of aaa. A cache evaluation system prevents a full scan to be performed for each loop: it is also possible to use many levels of joint-processing without lost of performance.

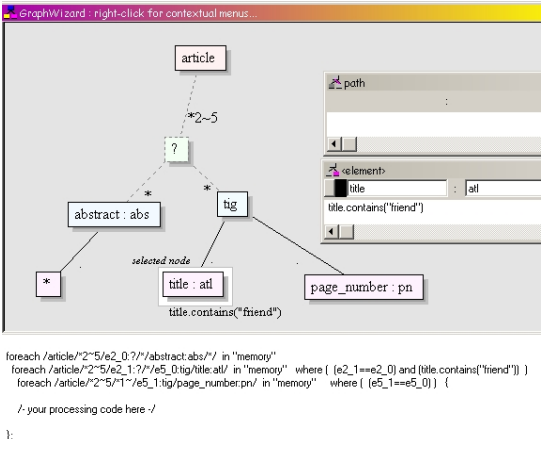


Fig. 1. Graph editor that generates an *alphaScript* request

In the example above, only the "idxmem" index database is used (see *in "idxmem"*), but each `foreach` doesn't not have to be performed on the same index database: it is therefore possible to perform joint on different index databases, unlike most actual XML query languages. This can be used to split or merge different index databases to merge or produce new XML documents.

A tree matching process featuring don't care characters is even possible using a nested collection of *foreach* linked by variables (Fig.1):

The XML Wikipedia [10] and the INEX2005 DB has been used to evaluate the system. The index size is 6.5 times the size of the XML collection. The indexing speed is of 3Mb/mn of source data on a P4 3 GHz. For each node and path, an entry in a QDBM hashtable on hard drive is created and a defined set of $S_{prop\_node}(N)$ (respectively $S_{prop\_path}(P)$) is added. The signatures of the paths are computed and inverted lists of path references are created using the QDBM's depot model. For precise requests (ie – with no *) and requests of a length of 8, 87% of $SC_{path}(C)$ is discarded by the signature's test for $MaxP_{distance}=0$. Of course, for 'very' imprecise requests (for instance '*'), no $P_{candidate}$ should be discarded (as awaited).

For the backtrack computations and paths of length n, experimentally we found that the mean operation's number stays under 5.n (under 40 cell visits for a pattern/path mean length of 8): that remains quite acceptable in regards to the matrix computation complexity.

Since a request can provide a large amount of solutions and even enumerate all the paths in the database (see '`foreach a:*`' for instance), we implemented an asynchronous interface system to provide the answers to the user as soon as they are available. The mean processing speed is of 2500 nodes evaluated per seconds.

The answer speed depends of course on the complexity of the request and processing. When no processing is involved, a maximal distance of 0, and no don't care characters used (such as '`foreach /article/chapter/b:text/ where (b.containsWord("travel")) {}`'), the system returns the answers within a tenth of second. The use of length undefined 'don't care' symbols slows down the search, as well as the use of a maximum accepted distance. For instance, the use of a path of length 10 with 2 '*' (no min/max) and with a max distance of 2 leads to a mean answer time of 4s on the whole database (between 2s and 6s). Using min/max for the don't care characters speeds up the search, because more signatures are rejected – and more path candidates as well.

6 Conclusion

We focused on a mechanism based on a dedicated editing distance and a t-uple matching process to address the matching of imprecise paths. This computation is used in a scripting language (*alphaScript*) to retrieve DOM node references in an approximate way. The assignment of the t-uple of variables is used to evaluate condition and to perform joints filtering in the same index database or even between two (or more) databases. This crafts one of many iterators used in the script language of alpha, a system designed to manage a large collection of XML documents.

References

1. Akutsu, T.: Approximate String Matching with Variable Length Don't Care Characters. IEICE Trans. E78-D (1996)
2. Amer-Yahia, S., Lakshmanan, L.V.T., Pandit, S.: FlexPath: Flexible Structure and Full-text Querying for XML
3. Amer-Yahia, S., Botev, C., Shanmugasundaram, J.: TeXQuery: A Full-Text Search Extension to XQuery in WWW 2004 (2004)
4. Baeza-Yates, B., Navarro, G.: XQL and Proximal nodes. In: Proceedings ACM SIGIR 2000 Workshop on XML and Information Retrieval (2000)
5. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, Reading (1999)
6. Botev, C., Amer-Yahia, S., Shanmugasundaram, J.: A TexQuery-Based XML Full-Text Search Engine (Demo Paper). In: SIGMOD (2004)
7. Bremer, J., Gertz, M.: XQuery/IR: Integrating XML Document and Data Retrieval. In: WebDB 2002, pp. 1–6 (2002)
8. Chinenyanga, T., Kushmerick, N.: An expressive and Efficient Language for Information Retrieval. JASIST 53(6), 438–453 (2002)
9. Cole, R., Gottlieb, L., Lewenstein, M.: Dictionary Matching and Indexing With Errors and Don't Cares. In: Annual ACM Symposium on Theory of Computing. Proceedings of the thirty-sixth annual ACM symposium on Theory of computing, vol. 2B, pp. 91–100 (2004)
10. Denoyer, L., Gallinari, P.: The Wikipedia XML Corpus, SIGIR Forum (2006)
11. Fisher, M., Patterson, M.: String Matching and Other Products. Complexity of Computation, SIAM-ACM Proceeding 7, 113–125 (1974)
12. Fragment Description, `http://www.w3.org/TR/WD-xml-fragment#terminology`
13. Fuhr, N., Grojohann, K.: XIRQL: A Query Language for Information Retrieval in XML Documents. In: SIGIR 2001 (2001)
14. Guo, L., Shao, F., Botev, C., Shanmugasundaram, J.: XRANK: Ranked keyword search over XML Documents. In: SIGMOD, pp. 16–27 (2003)
15. Landau, G.M., Vishkin, U.: Fast parallel and serial approximate string matching. Journal of Algorithm 10, 157–169 (1989)
16. Levenshtein: Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Doklady Akademii Nauk SSSR 10(8), 707–710 (1966) (Russian) (English translation in Soviet Physics Doklady (aka report) 10(8), 707–710 (1966))
17. Manber, U., Baeza-Yates, R.: An Algorithm for String Matching with a Sequence of Don't Cares. Information Processing Letters 37, 133–136 (1991)
18. Ménier, G., Marteau, P.F.: Information Retrieval in Heterogeneous XML Knowledge Bases. In: The 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Annecy, France, July 1-5. IEEE, Los Alamitos (2002)
19. Meyer, H., Bruder, I., Weber, G., Heuer, A.: The Xircus Search Engine (2003), `http://www.xircus.de`
20. Myers, E.W., Miller, W.: Approximate Matching of Regular Expressions. Bulletin of Mathematical Biology 51, 5–37 (1989)
21. Popovici, E., Ménier, G., Marteau, P.-F.: SIRIUS: A lightweight XML indexing and approximate search system at INEX 2005. In: Fuhr, N., Lalmas, M., Malik, S., Kazai, G. (eds.) INEX 2005. LNCS, vol. 3977, pp. 321–335. Springer, Heidelberg (2006)
22. Prolog Development Center (PDC), `http://www.pdc.dk`

23. Theobald, A., Weikum, G.: Adding Relevance to XML. In: WebDB 2000 (2000)
24. Theobald, A., Weikum, G.: The index-based XXL search engine for querying XML data with relevance ranking. In: Jensen, C.S., Jeffery, K., Pokorný, J., Šaltenis, S., Bertino, E., Böhm, K., Jarke, M. (eds.) EDBT 2002. LNCS, vol. 2287, pp. 477–495. Springer, Heidelberg (2002)
25. Trotman, A., Sigurbjörnsson, B.: Narrowed Extended XPath I (NEXI). In: Pre-Proceedings of the INEX 2004 Workshop. Schloss Dagstuhl, Germany, pp. 219–227 (2004)
26. QDBM sourceforge: M. Hirabayashi, http://qdbm.sourceforge.net/
27. Visual Prolog, product, http://www.visual-prolog.com
28. Wang, H., Park, S., Fan, W., Yu, P.: ViST: A Dynamic Index Method for Querying XML Data by Tree Structures. In: SIGMOD (2003)
29. XML-QL, http://www.w3.org/TR/NOTE-xml-ql/
30. XPath Reference, http://www.w3.org/TR/xpath
31. XQuery Reference: A Query Language for XML (Feburary 2001), http://www.w3.org/TR/xquery
32. XQuery & Exist, http://exist.sourceforge.net/
33. Zhang, K., Shasha, D., Wang, J.: Approximate Tree Matching in the Presence of Variable Length Don't Cares". Journal of Algorithms 16, 33–66 (1994)

A Comparison of XML-Based Temporal Models

Khadija Abied Ali[1] and Jaroslav Pokorný[2]

[1] Sebha University, Faculty of Science, Sebha, Libya
Khadijaabied91@yahoo.co.uk
[2] Charles University, Faculty of Mathematics and Physics, Praha , Czech Republic
jaroslav.pokorny@mff.cuni.cz

Abstract. Much research work has recently focused on the problem of representing historical information in XML. This paper describes a number of temporal XML data models and provides their comparison according to the following properties: time dimension (valid time, transaction time), support of temporal elements and attributes, querying possibilities, association to XML Schema/DTD, and influence on XML syntax. We conclude that the approaches of managing time information using XML mostly do not require changes of current standards.

Keywords: XML, temporal XML data model, bitemporal XML data model, versioning XML documents, transaction time, valid time, efficacy time, native XML databases (NXDs).

1 Introduction

Recently, the amount of data available in XML [13] has been rapidly increasing. In context of databases, XML is also a new database model serving as a powerful tool for approaching semistructured data. Similarly to relational or object-relational models in the past, database practice with XML started to change also towards using time in some applications, such as banking, inventory control, health-care, and geographical information systems. Much research work has recently focused on adding temporal features to XML, i.e. to take into account change, versioning, evolution and also explicit temporal aspects of XML data, like, e.g., the problem of representing historical information in XML. XML documents are related to time in two aspects: they contain temporal information and/or their contents evolve with time. Examples of the latter case include normative texts, product catalogues, etc. We consider both aspects in the temporal XML data models.

To manage temporal information in XML documents, a temporal XML data model is required. Based on similar approaches well-known from the field of temporal relational databases [9], many researchers transformed old ideas into the world of hierarchical structures of XML documents. Technically, to develop an XML temporal data model, it is necessary to extend a XML data model by a time dimension. The problem is that there is more XML data models (e.g. Infoset [14], XPath data model [3], XQuery data model [1], etc.) and more times (usually valid and transaction times).

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 339–350, 2009.

This fact complicates a comparison of various temporal XML data models that occur in literature.

In [9] the relational temporal data models are classified as two main categories: temporally ungrouped models and temporally grouped data models. As opposed to the former, a *temporally grouped* data model can be expressed by relations in non-first-normal-form model or attribute time stamping, in which the domain of each attribute is extended to include the temporal dimension. The hierarchical structure of XML provides a natural environment for use of temporally grouped data models. We describe a number of temporally grouped XML data models and provide their comparison according to the following properties: time dimension (valid time, transaction time), support of temporal elements and attributes, querying possibilities (particularly in languages XPath and XQuery), association to XML Schema/DTD, and influence on XML syntax.

The paper is organized as follows. Section 2 contains a brief overview of some works which have made important contributions on adding temporal features to XML. We describe an XML-based bitemporal data model (XBIT) and its application for versioned documents. Then we describe a temporal XML data model which is able to manage the dynamics of normative texts in time. We introduce also a valid-time XPath data model. The model adds valid time support to XPath language. Then we present a key-based approach for archiving data. The last model introduced is a multidimensional XML model. In Section 3 we summarize all the mentioned models. We briefly analyze their characteristics and subsequently we express our point of view. In Section 4, we describe shortly an ongoing work of a new XML-based temporal model. Finally, in Section 5, we present our conclusions and future investigations.

2 Time in XML – an Overview

As usually, an XML data model should provide tools for describing structure of XML data, integrity constraints, manipulation statements, and querying XML data. We mostly omit integrity constraints and in manipulations we focus on promitive change operators for elements and/or attributes: update, insert, and delete.

2.1 XBIT - an XML-Based Bitemporal Data Model

The approach introduced in [11] is based on temporally-grouped data model. A temporal XML document is represented by adding two extra attributes, namely `vstart` and `vend`, representing the time interval for which an element is valid. `vstart` and `vend` represent the inclusive valid time interval (`vend` can be set to the special symbol `now`) to denote the ever-increasing current date. Each temporal element is assigned also two extra attributes; `tstart` and `tend` to represent the inclusive transaction time interval; `tend` can be set to UC (until changed).

Example 1: A temporal element `title` can be represented in XBIT in the following way:

```
<title vstart="1998-01-01" vend="now"
       tstart="1997-09-01" tend="UC">Sr Engineer</title>
```

The expression says that `title` with the value `Sr Engineer` is valid from 1998-01-01 until now. This fact is recorded in the database in 1997-09-01.

The model can also support powerful temporal queries expressed in XQuery without requiring the introduction of new constructs in the language; all the constructs needed for temporal queries can be introduced as user-defined libraries. Modifications in XBIT can be seen as the combination of modification on valid time and transaction time history. XBIT will automatically *coalesce* on both valid time and transaction time, for instance, the valid time intervals with the same title value that overlap or are adjacent. Those intervals are merged by extending the earlier interval; this process is repeated until maximal intervals are constructed. At this point, the non-maximal intervals are removed. The used technique is general and can be applied to historical representation of relational data and versions management in archives (see Section 2.2).

2.2 An XML–Based Model for Versioned Documents

An efficient technique for managing multiversion document histories [10] is used by storing the successive versions of a document in an incremental fashion. Elements of an XML document use again attributes `vstart` and `vend` representing the time interval for which this elements version is valid. Elements containing attributes can be supported by representing each attribute as a subelement denoted by a special flag attribute `isAttr`. For instance, if the employee element contains the attribute `empno` then this fact is represented as:

```
<empno isAttr="yes" vstart="1999-01-01" vend="now"> e1
</empno>
```

Change operations on XML documents can be represented in the model for elements. Three primitive change operations are considered, delete, insert, and update. The following is the effect of performing each operation:

Update. When the element *a* is updated at time *t*:

1. a new element with the same name *a* will be appended immediately after the original one; the attributes *vstart* and *vend* of this new element are set to *t* and *now*, respectively.
2. the *vend* attribute of the old element is set to *t* – 1.

Consider the element `<a vstart="2000-01-01" vend="now">123</a>`, and suppose to change the value of *a* to `250` at `"2002-01-02"`. Then we get the following elements:

```
<a vstart="2000-01-01" vend="2002-01-01">123</a>
<a vstart="2002-01-02" vend="now">250</a>
```

Insert. When a new element is inserted at time *t*, this element is inserted into the corresponding position in the document; the *vstart* attribute is set to *t*, and *vend* is set to *now*.

Delete. When an element is removed at time *t*, the *vend* attribute is set to *t* -1. Suppose to delete the element *a* at 2004-01-02. Then we get the following element:

```
<a vstart="2002-01-02" vend="2004-01-01">250</a>
```

The DTD of the versioned XML document can be automatically generated from the original DTD. Two new attributes `vstart` and `vend` are added to each element; an attribute of an element will be converted as a child element. For instance, the temporal element `title` is represented in the DTD as

```
<!ELEMENT title (#PCDATA)>
<!ATTLIST title vstart CDATA #REQUIRED
                vend CDATA #REQUIRED>
```

Due to the temporally grouped features of the model, it is possible to express powerful temporal queries in XQuery. For instance, the query "Find chapters in which the title did not change until a new History section was added" can be expressed as

```
for $ch in document ("V-Document.xml")/document/chapter
let $title:= $ch/title[1]
let $sec:= $ch/section[. ="History"]
where not empty($title) and not empty($sec) and
      $title/@vend = $sec/@vstart
return $ch
```

2.3 An XML-Based Temporal Data Model for the Management of Versioned Normative Texts

In this model [7], four dimensions (publication, validity, efficacy, and transaction times) are used in the context of legal documents. A temporal element is chosen as a basic unit of temporal pertinence. They are used to represent the evolution of norms in time and their resulting versioning. *Publication time* is the time of publication of norms in an official journal. *Efficacy time* usually corresponds to the validity of norms, but sometimes the cancelled norm continues to be applicable to a limited number of cases. *Valid time* represents the time the norm is in force (the time the norm actually belongs to the regulations in the real world). *Transaction time* is the time the norm is stored in the computer system.

An alternative XML encoding schema has been developed for normative text based on an XML-schema which allows the introduction of time stamping metadata at each level of the document structure which is a subject to change. For the management of norms, three basic operators are defined; one for the reconstruction of a consistent temporal version and the other two for the management of textual and temporal changes. Querying uses combination of full text retrieval and XQuery extended by some constructs to deal with time dimensions.

2.4 A Valid Time XPath Data Model

In the valid time XPath data model [12] a list of disjoint intervals or instants that represent the valid time is added to each node of the original XPath data model.

- Every node of the tree structure of an XML document is associated with the valid time that represents when the node is valid, no node can exist at a valid time when its parent node is not valid.

- The valid time of any node is a superset of the union of the valid times of all its children as well as all its descendents. The valid time of the root node should be a superset of the union of the valid times of all nodes in the document.
- The valid time of an edge is determined by the valid time of the nodes at the edge's two ends (if both nodes are valid, an edge can exist). The valid time of the edge is result of $t_1 \cap t_2$, where t_1 and t_2 are the valid times of the edge's two ends.

The XPath is extended with an axis to locate the valid time of a node. A valid time axis is added to XPath to retrieve nodes in a view of the valid time for a node. The axis returns a list of nodes relative to the context node. Each node in an XML document has a corresponding valid time view containing its valid time information. Here, a valid time list can be viewed as an XML document. Each time in the valid time list is denoted as a <time> element. The content of the <time> is unique to the view. Below we show the valid time view of an element in the commonly used Gregorian calendar; "year", "month ", and "day" element nodes are nested under "begin" and "end" of each view.

```
<validtime>
    <time>
      <begin>
        <day>31</day>
        <month>Jan</month>
        <year>1999</year>
      </begin>
      <end>
        <day>now</day>
        <month>now</month>
        <year>now</year>
      </end>
    </time>
</validtime>
```

Remind that any calendar can be used. The commonly used calendar is Gregorian calendar; however there are other calendars that are widely used by people in different regions.

The valid time axis selects the list of nodes that form a document-order traversal of the valid time view. By this constraint, the nodes in the valid time axis are ordered according to the document order traversal of the valid time view. The valid time axis of a node contains the valid time information of the node as if it had originated from an XML document (a document order refers to the standard document order as it is specified in Infoset).

Since the <time> elements in the valid time view are ordered by the actual time they represent, these <time> elements selected by the valid time axis are also in this order.

Example 2: Below are some simple examples of using the valid time axis to query within the default view of the valid time.

`v/valid`	specifies the valid time axis of the node *v*.
`v/valid::day`	selects all the day nodes in the axis.
`v/valid::time[2]`	selects the second time node in the axis .

2.5 Key-Based Model for Archiving Data

In this archiving technique [2], a document is viewed as unordered set of XML elements. The elements can be uniquely identified and retrieved by their logical keys; elements have timestamps only if they are different from the parent node.

Key-based approach is used for identifying the correspondence and changes between two given versions based on keys. In contrast to diff-based approach which (i) keeps a record of changes – a "delta" - between every pairs of consecutive versions, (ii) stores the latest version together with all forward completed deltas –changes between successive versions- that can allow one to get to an earlier version by inverting deltas on the latest version, the Key-based approach can preserve semantic continuity of each data element in the archive. An element may appear in many versions whose occurrences are identified by using the key structure and store it only once in the merged hierarchy.

A *key* is a pair $(Q, \{P_1,\ldots, P_k\})$ where Q and P_i, $i \in [1, k]$, are path expressions in a syntax similar to XPath. Informally, Q identifies a target set of nodes reachable from some context node and this target set of nodes satisfies the key constraints given by the paths, P_i, $i \in [1, k]$.

Example 4: Consider the XML document

```
<DB>
   <A> <B>1</B> <C>1</C> </A>
   <A> <B>1</B> <C>2</C> </A>
</DB>
```

The document satisfies the key `(/DB/A,{C})` but it does not satisfy the key `(/DB/A,{B})` since both `A` elements have the same key path value, i.e., `<B>1</B>`.

This archiving technique requires that all versions of the database must conform to the same key structure and the same schema as well.

All the versions are merged into one hierarchy where an element appearing in multiple versions is stored only once along with a timestamp. The main idea behind nested merge is:

- Recursively to merge nodes in D (incoming version) to nodes in A (the archive) that have the same key value, starting from the root.
- When a node y from D is merged with a node x from A, the time stamp of x is augmented with i (the new version number).The sub-trees of nodes x and y are then recursively merged together.
- Nodes in D that do not have corresponding nodes in A are simply added to A with the new version number as its time stamp.
- Nodes in A that no longer exist in the current version D will have their timestamps terminated appropriately, i.e., these nodes do not contain timestamp i.

Since the archive is in XML, the existing XML query languages such as XQuery can be used to query such documents. The authors of [2] did not discuss the issue of temporal queries in detail.

2.6 A Multidimensional XML Model (MXML)

In the approach [5], we represent multiple versioning not only with respect to time but also to other context parameters such as language, degree of detail, etc.

In a multidimensional XML document, dimensions may be applied to elements and attributes. A multidimensional element/attribute is an element/attribute whose contents depend on one or more dimensions. The notion of *world* is fundamental in MXML. A *world* represents an environment under which data in a multidimensional document obtain a meaning. A *world* is determined by assigning values to a set of dimensions.

Example 5: Consider the *world w* = {(*time*, 2005-12-14), (*customer_type*, student), (*edition*, English)}. The dimensions names are *time*, *customer_type,* and *edition*. The assigned values of these dimensions names are 2005-12-14, student, and English respectively.

The multidimensional element is denoted by preceding the element's name with the special symbol **"@",** and encloses one or more context elements. All context elements of a multidimensional element have the same name which is the name of the multidimensional element. Consider the following MXML document:

```
<book>
  <@isbn>
   [edition = greek]  <isbn>0-13-110370-9</isbn> [/]
   [edition = English] <isbn>0-13-110362-8</isbn> [/]
  </@isbn>
</book>
```

The `@isbn` is a multidimensional element dependent on the dimension `edition`. It has two context elements having the same name `isbn` (without the special symbol "@"). Context specifiers qualify the facets of multidimensional elements and attributes, called *context elements/attributes*, stating the sets of worlds under which each facet may hold. `[edition = greek]` and `[edition = English]` are the context specifiers of `@isbn`. [/] represents the end symbol of a context specifier.

Change operations (update, delete, insert) can be represented in MXML for both elements and attributes. For instance, consider the element `<p` a_1 `="9">` v_1 `</p>` and suppose to delete the attribute a_1 at time point `t`. Then we get the following MXML element:

```
<p a1=[d in{start..t-1}]"9"[/]>v1</p>
```

where a dimension named `d` is used to represent time, `start` is a reserved word representing the start time, `t-1` represents the end time.

The history of the schema of an XML document can be represented easily, for instance, deleting an element, or adding an attribute to an element at a specific time point. The following XML schema description retain the element `r` as optional during the interval `{t..now}`.

```
<xs:element name="r" type="xs:string"
            minOccurs=[d in {t..now}]"0"[/]/>
```

Consider the following XML document:

```
<p>
   <q>v1</q> <r>v2</r> <s>v3</s>
</p>
```

The schema for this document may be encoded in XML schema as follows:

```
<xs:element name="p">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="q" type="xs:string"/>
         <xs:element name="r" type="xs:string"/>
         <xs:element name="s" type="xs:string"/>
      </xs:sequence>
   </xs:complexType>
</xs:element>
```

Suppose to delete the element `r` at time point `t`. Then we get the following MXML element:

```
<p>
   <q>v1</q>
   <@r>
      [d in {start..t-1}] <r>v2</r> [/]
   </@r>
   <s>v3</s>
</p>
```

After deleting the element `<r>v2</r>` at time `t`, it is necessary to modify the document's schema if we want the XML document resulting by applying the deletion to become valid. This change can be represented by turning the element sequence of the above XML schema into a multidimensional element with two facets:

```
<xs:element name="p">
   <xs:complexType>
      <@xs:sequence>
         [d in {start..t-1}]
            <xs:sequence>
               <xs:element name="q" type="xs:string"/>
               <xs:element name="r" type="xs:string"/>
               <xs:element name="s" type="xs:string"/>
            </xs:sequence>
         [/]
         [d in {t..now}]
            <xs:sequence>
               <xs:element name="q" type="xs:string"/>
               <xs:element name="s" type="xs:string"/>
            </xs:sequence>
         [/]
      </@xs:sequence>
   </xs:complexType>
</xs:element>
```

3 Summary of XML-Based Temporal Data Models

So far, we have introduced some works which have made important contributions in providing expressive and efficient means to model, store, and query XML-based temporal data models. In the following subsections we provide a comparison of all mentioned models according to the following properties: time dimension (valid time, transaction time), support of temporal elements and attributes, querying possibilities, association to XML Schema/DTD, and influence on XML syntax.

Time dimension. All the models are capable to represent changes in an XML document by supporting temporal elements, and incorporating time dimensions. Two time dimensions are usually considered: valid time and transaction time. There are several other temporal dimensions that have been also mentioned in the literature in relation to XML. In [7] a publication time and efficiency time in the context of legal documents are proposed.

Temporal elements and attributes. Time dimensions may be applied to elements and attributes. All the models are capable to support temporal elements. In [5] and [10] the temporal attributes are supported. In our point of view, supporting temporal attributes adds an advantage to the model. In [5] versions of an element are explicitly associated as being facets of the same (multidimensional) element. Grouping facets together allows the formulation of cross-world queries, which relate facets that hold under different worlds [6].

Influence on XML syntax. Temporal information is supported in XML much better than relational tables. This property is attributed to the hierarchical structure of XML which is compatible perfectly with the structure of temporal data. Only in [5] the syntax of XML is extended in order to incorporate not only time dimensions but also other dimensions such as language, degree of detail, etc. So the approach in [5] is more general than other approaches as it allows the treatment of multiple dimensions in a uniform manner.

Querying possibilities. In our point of view, the model's power depends also on supporting powerful temporal queries. In [10] and [11] powerful temporal queries expressed in XQuery without requiring the introduction of new constructs in the language are supported. In [12] a valid time support is added to XPath. This support results in an extended data model and query language. In [7] querying uses combination of full text retrieval and XQuery extended by some constructs to deal with time dimensions. The other models in [5] and [2] did not discuss the issue of temporal queries; in [2] elements have timestamps if they are different from the parent nodes. This fact complicates the task of writing queries in XPath/XQuery**;** consider the following XML representation of an archive containing versions 1 and 2.

```
<T t="1-2">
   <db>
      <employee>
         <id>1</id>
         <name>Anas</name>
         <address>
            <city>Prague</city>
            <street>Krouzova 18</street>
```

```
            </address>
            <salary><T t="1">22k</T>
                    <T t="2">30k</T></salary>
        </employee>
        ...
    </db>
</T>
```

Note that `T` is a special element represents element's versions by a special attribute `t`. For instance, the tenth line says that the value of `salary` is `22k` and `30k` in the first and second version respectively. The salary of `Anas` in the second version can be expressed easily in XQuery (because salary elements have their own timestamps):

```
//employee[name="Anas"]/salary/T[@t="2"]
```

But where `Anas` was living in the second version can not be expressed easily in XQuery. It requires to check timestamps of `db` element since `city`, its parent (`address`), and parent of its parent (`employee`) do not have timestamps.

Association to XML Schema/DTD. A significant advantage will be added to the model if it is not only representing the history of an XML document but also the history of its corresponding XML schema or DTD as well. In [5], [7], and [10] the temporal XML schema/DTD is supported by extending the existing XML schema/DTD. All the mentioned models are summarized in Table 1.

4 3D_XML: A Three-Dimensional XML-Based Model

In this Section we describe shortly an ongoing work of a new XML-based temporal model. Our model is a three-dimensional XML-based model (3D_XML in short) for representing and querying histories of XML documents. The proposed model incorporates three time dimensions, *valid time*, *transaction time*, and *efficacy time* without extending the syntax of XML. An important issue of each data model is its implementation. Native XML databases *NXDs* represent a suitable storage platform when complex time dependent data has to be manipulated and stored, so we chose to implement temporal queries directly in NXDs (particularly *eXist*). 3D_XML is equipped with a set of temporal constructs; valid/efficient times relationships constructs, interval comparison operators, snapshot data construct, etc. We use XQuery to express complex temporal queries, but the expression of these queries is greatly simplified by a suitable library of built-in temporal functions; the preliminary experimental results on query performance are encouraging.

5 Conclusions

In this paper, we showed that an effective temporal information system should provide (i) expressive temporal data model, (ii) powerful language for temporal queries and snapshot queries as well. We conclude that XML provides a flexible mechanism to represent complex temporal data. Its query language XQuery is natively extensible and Turing-complete [8], and thus any extensions needed for temporal queries can be

defined in the language itself. This property distinguishes XML temporal querying from that one in relational temporal languages, e.g. TSQL. So, any syntax extension of XQuery towards a temporalness, e.g. τXQuery [4], makes only queries easier to write. The most usual bitemporal representation does not require changes of current XML standards.

Table 1. Features summary of XML-based temporal models. ("-" means that the associated feature is not discussed in the original model; Y and N denote Yes and No, respectively.

Model	Time dimension	Supports temporal elements	Supports temporal attributes	Querying	Extends XML schema/DTD	Extends XML syntax
XBIT [11]	Valid/ transaction time	Y	N	XQuery	-	N
Versioned Documents [10]	Valid/ transaction time	Y	Y	XQuery	Y	N
Versioned normative texts [7]	Valid/ transaction/ publication/ efficacy time	Y	N	XQuery is extended	Y	N
Valid time XPath data model [12]	Valid time	Y	N	XPath is extended	-	N
Key-based model for archiving data [2]	Valid time	Y	N	-	-	N
MXML [5]	Valid/ transaction time	Y	Y	-	Y	Y

XML can be even an option for implementations of temporal databases (or multidimensional databases) on a top of a native XML DBMS. Our work shows that there are a lot of important topics for forthcoming research. Many research issues remain open at the implementation level, including, e.g., the use of nested relations on the top of an object-relational DBMS, reflecting temporal features into an associated XML query language, etc. The future work is directed to extend 3D_XML model by adding more temporal constructs in order to support more powerful temporal queries. Support of updates will be also a real area of future investigations.

Acknowledgements. The research was in part supported by grant 1ET100300419 of the Information Society Program - Thematic Program II of the National Research Program of the Czech Republic.

References

1. Boag, S., Chamberlin, D., Fernández, M.F., Florescu, D., Robie, J., Siméon, J.: XQuery 1.0: An XML Query Language, W3C Working Draft, 04 (April 2005), http://www.w3.org/TR/xquery/
2. Buneman, P., Khanna, S., Tajima, K., Tan, W.: Archiving scientific data. In: Proc. of ACM SIGMOD Int. Conference, pp. 1–12 (2002)
3. Clark, J., DeRose, S.: XML Path Language (XPath) Version 1.0, W3C Recommendation (November 16, 1999), http://www.w3.org/TR/xpath/
4. Geo, D., Snodgrass, R.: Temporal slicing in the evaluation of XML queries. In: Proc. of VLDB, Berlin, Germany, pp. 632–643 (2003)
5. Gergatsoulis, M., Stavrakas, Y.: Representing Changes in XML Documents using Dimensions. In: Proc. of 1st Int. XML Database Symposium, pp. 208–221 (2003)
6. Gergatsoulis, M., Stavrakas, Y., Doulkeridis, C., Zafeiris, V.: Representing and querying histories of semistructured databases using multidimensional OEM. Inf. Syst. 29(6), 461–482 (2004)
7. Grandi, G., Mandreoli, F., Tiberio, P.: Temporal Modelling and Management of Normative Documents in XML Format. Data and Knowledge Engineering 54(3), 227–254 (2005)
8. Kepser, S.: A Simple Proof of the Turing-Completeness of XSLT and XQuery. In: Proc. of Extreme Markup Languages, Montréal, Québec (2004)
9. Tansel, A., Clifford, J., Gadia, S., Jajodia, S., Segev, A., Snodgrass, R.T.: Temporal Databases: Theory, Design and Implementation, pp. 496–507. Benjamin/Cummings Publishing Company, California (1993)
10. Wang, F., Zaniolo, C.: Temporal Queries in XML Document Archives and Web Warehouses. In: Proc. of 10th Int. Symposium on Temporal Representation and Reasoning, pp. 47–55 (2003)
11. Wang, F., Zaniolo, C.: XBIT: An XML-based Bitemporal Data Model. In: Proc. of 23rd Int. Conference on Conceptual Modeling, Shanghai, China, pp. 810–824 (2004)
12. Zhang, S., Dyreson, C.: Adding Valid Time to XPath. In: Proc. of 2nd int. Workshop on Database and Network Information Systems, Aizu, Japan, pp. 29–42 (2002)
13. W3C: Extensible Markup Language (XML) 1.1, 3rd edn., W3C Recommendation (February 04, 2004), http://www.w3.org/TR/xml11/
14. W3C: XML Information Set, 2nd edn., W3C Recommendation (February 04, 2004), http://www.w3.org/TR/xml-infoset/

An Adaptation Approach: Query Enrichment by User Profile

Corinne Amel Zayani[1,2], André Péninou[1,2], Marie-Françoise Canut[1,2], and Florence Sèdes[1,2]

[1] IRIT, 118 route de Narbonne, 31062 Toulouse cedex 4, France
zayani,sedes@irit.fr
http://www.irit.fr
[2] LGC, 129 A, avenue de Rangueil B.P 67701, 31077 Toulouse cedex 4, France
peninou, canut@iut-blagnac.fr

Abstract. In semi-structured information systems, generally, the adaptation of documents is essential to give the user the feeling that the query result is adapted to his preferences. The user's needs can be defined in a user profile. But, in the literature, adaptation systems are designed for a particular domain and are oriented towards either navigation adaptation or content adaptation. Adaptation takes place after the user's query has been evaluated. So, in this paper, we contribute to propose an adaptation algorithm which is domain independent and whose adaptation takes place before user's query evaluation. This algorithm consists in enriching the user query on the basis of user profile in order to adapt the results to the user.

1 Introduction

In semi-structured information systems, the adaptation of documents is defined for a particular domain [6], such as educational domain, e-commerce, tourism, etc. On the other hand, it depends on a user profile which aims to descibe some user characteristics [6]. The adaptation can be applied to the content or/and navigation, in order to answer respectively the problems of cognitive overload and disorientation [8]. The adaptation is very often supported by Event-Condition-Action rules [1] that are defined and implemented in adaptation engines. The engines trigger the rules over the attributes defined in the user model (e.g. number of visits, knowledge degree, etc.) and in the domain model. These rules can also update the user model that are expressed, for example, by logics of description.

The problem of adaptation within generic framework is recently tackled by [9]. In its thesis, [9] adapts the document presentation to particuler user's characteristics (blind user, normal user). The user defines explicitly his presentation preferences for each URL resource. We can then notice that adaptation systems are directly related to either a particular domain or to a particular user.

Within a generic framework and in order to use implicit user characteristics, we have yet proposed in [16] a domain independent architecture. This architecture extends those defined in the literature such as AHAM [4] or Munich reference model [12]. The particularity of this proposed architecture is that the

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 351–361, 2009.

domain model has been replaced by a document model that was defined by [2] and the user model is built from the analysis of user's queries. Moreover, two adaptation processes have been defined in the architecture: upstream adaptation and downstream adaptation.

In this paper, we are interested in explaining the upstream adaptation process. We suppose that we have a document repository containing documents which can belong to any domain. To query a document or the document collection, the user builds his query textually or graphically [5]. On the other hand, we suppose that we have a user profile repository that clusters the needs of each user. The aim of this upstream adaptation process is to reduce the problem of cognitive overload when result is presented to the user. In this purpose, we have proposed three sub-algorithm. The first algorithm aims to determine the user preferences for query enrichment. The second algorithm aims to enrich the user query by user preferences. The third algorithm aims to update the user profile by user query.

The paper is structured as follows. Next section presents a brief summary of adaptation systems. Section 3 describes user's characteristics of the user model. Section 4 presents the adaptation process. In section 5 the adaptation algorithm is defined. In section 6 we present an application. An evaluation is made on this application in section 7. Conclusion and future work close the paper.

2 Related Works

The AHAM (Adaptive Hypermedia Application Model) [4] model has been originally defined for educational domain. This model is based on the generic Dexter Model [10], but remains limited to applications in educational adaptive hypermedia systems. It splits up the storage layer of the Dexter Model into an adaptation model, a domain model, and a user model. The user model is an overlay of the domain model. The adaptation engine implements a set of adaptation rules for educational purpose. Like in AHAM, [12] have described the Munich reference model for adaptive hypermedia applications, where the adaptation engine can implement not only the educational-oriented rules but also other rules which can be defined for a particular domain by an expert. Like in this latter model, AHA! system [3] is defined for adaptive Web application. As explained earlier, AHAM, Munich reference model and AHA! perform rule-based adaptation. Consequently the adaptation engine only implements domain-based rules. Some works tried to reduce this restriction by defining non-persistent properties and post and pre concepts access rules execution. [14] have defined GAM that is a generic theoretical model for describing the user adaptive behaviour in a system in order to adapt interactive systems. This problem has been dealt with [9] when blind users interact with information systems.

These different architectures for adaptive hypermedia systems are oriented towards particular domains or interactive systems. So, we have yet proposed in [16] to change the domain model by a document model relative to any domain. In this proposal, the user profile can include different interests relative to any domain. It is built from the analysis of user's queries. Because of limitation of

content, we don't present the document model. The reader can refer to [16]. In this paper, we present the user profile and the adaptation process.

3 User Profile

As adaptation depends on the user model, we suggest that the user model should describe structure of each user profile. We suppose that user profile is structured in XML format. The user profile contains characteristics that can be distinguished into two types [6] [11]:

- *permanent characteristics* which are constant over time. This type of characteristics introduces the user identity (last name, first name, etc.). It is useful in order to identify the user,
- *changing characteristics* which evolve over time. This type of characteristics introduces the user preferences. We have proposed to automatically built user preferences from an analysis of his previous queries. It is useful in order to enrich user query and adapt the result. The structure of changing characteristics is the same as the structure of user query, which allows to make comparison easier between user profile, query and the document in order to enrich the query. The figure 1 shows the changing characteristics in XML schema.

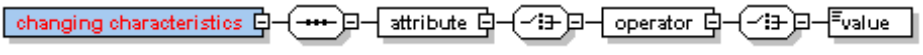


Fig. 1. changing characteristics

4 Adaptation Process

The adaptation process is the key issue in this paper: it describes how the adaptive hypermedia system should perform its adaptation. In the literature, adaptation process is defined as a set of adaptation rules that make connection between domain model and user model. But each set of adaptation rules is designed for a particular domain because they take into account the relationships defined between the different concepts of the domain. The rules are triggered when the user chooses a link in a document. These rules adapt either the content or the links (navigation) of the chosen document that must belong to the targeted domain (domain model).

But such rules don't perform any adaptation when the user queries a document collection. To tackle this problem, we have distinguished two adaptation processes respectively named : downstream and upstream.

- *The downstream adaptation process* is useful when the user browses in hypermedia infrastructure. The process is based on adaptation rules as described above. These rules are expressed for example by description logics. This process aims to reduce the problem of disorientation.
- *The upstream adaptation process* is used when the user queries the document repository. This process aims to reduce the problem of cognitive overload.

In this paper, we are interested in the upstream adaptation process. So, we present an algorithm which aims is to reduce cognitive overload when results are presented to the user.

5 Proposal of an Algorithm for Upstream Adaptation

In order to support the upstream adaptation process, we have proposed an algorithm that is split in three sub-algorithm. The first sub-algorithm determines the user preferences for query enrichment. The second sub-algorithm enriches the user query. The third sub-algorithm updates the user profile.

5.1 User Preferences

In order to extract the user preferences from user profile, the first sub-algorithm verifies the similarity between the query elements properties[1] and profile elements properties. For query element, it is necessary to find their properties from the DTD or XML Schema of XML document. If the similarity is higher than a threshold, the profile element can be added to the query. In order to simplify the reading of the algorithm we note user profile by UP, a profile element by PE, a query by Q, a query element by QE and a DTD of XML document by XMLDC. The sub-algorithm for determining user preferences is :

```
(1) program extraction
(2) variable
(3) List_preferences[]: list of preference (attribute, value, operator)
(4) Begin
(5)     Traverse Q
(6)    Traverse UP
(7)     calculate the name similarity between PE and QE
(8)     If name similarity > 0 then // equation (1), see below
(9)          Traverse XMLDC
(10)             Until find QE property
(11)             Save the QE property;
(12)             Traverse UP
(13)             Traverse XMLDC
(14)             Until find PE property
(15)             Save the PE property;
(16)             calculate parent similarity between PE and QE ;
(17)         If parent similarity > threshold then //see equation (2)
(18)              add PE in the List_preferences[];
(19)         EndTraverse UP
(20)    End If
(21)      EndTraverse UP
(22)    EndTraverse Q
(23) End.
```

[1] The properties of the element consist in name(e), attribute(e), parent(e) and e refers to element.

The similarity measurement between the user profile and the query is determined on the basis of their elements properties. In this paper, we have interested to the similarity measurement for the names of elements taken from [15], that is determined by the name matching of the two elements a and b as follows :

$$s_{name}(a,b) = \frac{N_{n(a)\cap n(b)}}{N_{n(a)\cap n(b)} + \alpha(a,b)N_{n(a)/n(b)} + (1-\alpha(a,b))N_{n(b)/n(a)}} \quad (1)$$

where $N_{n(a)\cap n(b)} =\mid name(a) \cap name(b) \mid$ returns 1 if the two names have a common string, else 0. $N_{n(a)/n(b)} =\mid name(a)/name(b) \mid$ returns 1 if a has a string name difference with b string name, else 0.

For example, the two names "depart" and "department" have a common string "depart" and the set difference of two names is the string "ment". Therefore, $N_{department\cap depart} = 1, N_{department/depart} = 1, N_{depart/department} = 0$. Supposed that weight α =0.5 (the relative importance of $N_{n(a)/n(b)}$ and $N_{n(b)/n(a)}$ is the same), the similarity of two names is equal to $s_{name}(department, depart) = 0.57$.

This equation is used in the first part of the enrichment algorithm, in order to verify if the name of element query is similar to the name of an element profile.

In our context, we have defined the similarity of parents for a and b as follows :

$$s_{parent}(a,b) = \begin{cases} 1 \text{ if } s_{name}(parent(a), parent(b)) > 0 \\ 0 \text{ if } s_{name}(parent(a), parent(b)) = 0 \end{cases} \quad (2)$$

5.2 Query Enrichment

We have proposed another sub-algorithm which enables to enrich the query with user preferences, extracted through the previous algorithm. The goal is that the set of returned documents from collections should be the same (content and length) as if no adaptation was made, that is the set returned by the initial query. Our hypothesis of enrichment consists in ordering the documentary units in priority according to user preferences, in order to reduce the cognitive overload for the user.

In [16] we have proposed an algorithm that contains two generic functions which take into account only one user preference. So, we propose to improve the algorithm defined in [16] by the taking into account of "n" preferences. The enrichment of the query is made stage by stage. Each stage leads to a new "partial" query, that we can consider as a part of the adapted query to be evaluated. At each stage, two parts are inserted in the query being built: a static part and an evolutionary part. The first part is the conditions of the initial query. The second part comprises the elements extracted from the user profile (see section 5.1). The evolution of this part depends on the change of operators (e.g. $=$, $\leq$, $\geq$, etc) by their negation.

In the first stage, we keep the initial operators which are extracted from the profile. That corresponds to find all documents existing in the collection that both match the query and user preferences. Afterwards, in each stage, an operator of added user preferences will be replaced by its negation. The negation

of the operators is made in the increasing order of frequency existing in the user profile. That corresponds to find documents from the collection that match the query and match the less and less the user preferences: these documents first match all user preferences but not the lower frequent, and so on.

In the last stage, all operators for user preferences are the negation in comparison with the first stage. That corresponds to find documents that match the query but that not match user preferences.

Computing all possible and correct combinations, the number of query enrichment stages is equal to 2^n, where "n" is the number of the user preferences that will be extracted from the user profile (and added to the query). So, in order to adapt the results to the user, the system should evaluate these queries, in the order they have been generated. That ensures a correct ordering of the documents according to user's preferences. To set up these combinations, we use a matrix M [Line, Column] which has a number of lines (Line) equals to 2^n and has a number of columns (Column) equals to the preferences number. This matrix is filled by values 1 to indicate a user preference as stated in the user profile, and 0 to indicate the negation of the user preference. To produce this matrix, we use "combination_Preference" function based on classical binary combinations of n elements.

```
(1) program enrichment
(2) variable
(3)     List_preferences []: list of preference (attribute, value, operator)
(4) Begin
(5)     Matrix_Combination[2**nb_pref][nb_pref]:
            matrix for Preference combination
(6)     Sort List_preferences[] according to increasing
           order of frequences of user perferences ;
(7)     combination_Preference(Matrix_Combination [][]);
(8)     For each line i of Matrix_Combination
(9)         enriched Q := Q + user preferences of the current line of Matrix_Combination
                    with initial operator (1) or negation operator (0);
(10)         liste_query [i] := enriched Q;
(11)    End For
(12)    return liste_query;
(13) End.
```

This high level algorithm is implemented by generating XQuery expressions evaluated by an XQuery engine.

5.3 Update User Profile

We have proposed an algoritm in order to update the user profile and to manage frequencies of user preferences. It verifies if the properties (name or parent) of query elements are similar to the properties of profile elements (see section 5.1: equation 1 and 2). If it is the case, the element frequency in the profile is incremented. Else, the element will be added to the profile.

```
(1)program update
(2) Traverse UP
(3)   If PE is added in Q according to the previous algorithm then
```

```
(4)     PE.frequency++
(5)   Else
(6)     Traverse Q
(7)       Calculate the name similarity between PE and QE;
(8)       If name similarity > threshold then
(9)         PE. frequency++;
(10)          If condition exists in UP then
(11)           EP.condition.freqVal++;
(12)      Else
(13)        Add condition in UP;
(14)     EndTraverse Q
(15) EndTraverse UP
```

6 Application

In order to experiment our algorithm, we use a collection of XML documents of cottage rentings. Each document describes a rented cottage with structured data (number of beds, person number, etc.) and also raw text (rooms description, leisure activities, etc.). This collection exists in the PRETI platform[2] that allows the user to retrieve information according to several querying mechanisms, such as by user-defined preferences, by flexible operator, etc.

Our application aims to improve this platform by adding a repository for user profiles and adding the enrichement algorithm in order to adapt automatically the query results to the user. The figure 1 presents the XML Schema of cottage rentings document structure.

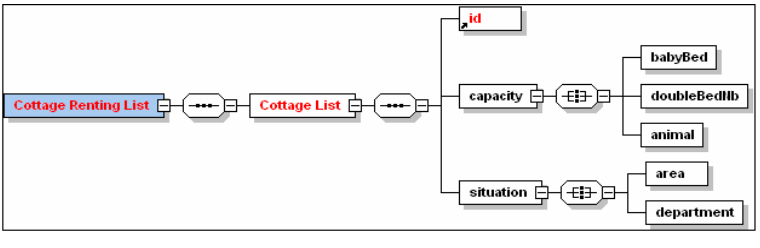


Fig. 2. Example of XML document of cottage renting

We suppose that a given user profile is presented on the right of the figure 3 and that the current query is presented on the left of the figure 3. The query is: find rented cottage in "aude" department accepting animals (department = "aude" and animal = "yes"). We have applied the algorithm proposed in the section 5. According to sub-algorithm of user preferences (see section 5.1) and document structure of figure 2, the figure 3 shows the elements which have similar properties by plain lines and broken lines groups.

[2] http://www.irit.fr/PRETI

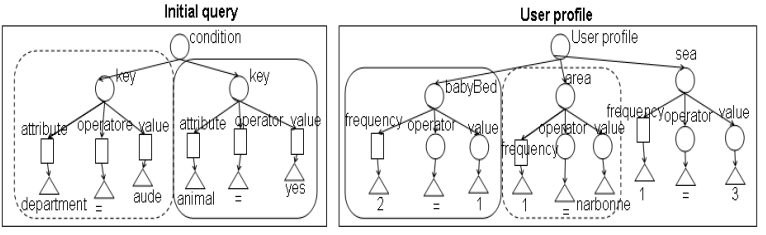


Fig. 3. Example of user preferences for query enrichment

According to sub-algorithm of query enrichment (see section 5.2), the resulting enriched query is presented in the form of four queries as follow (one query by stage of enrichment) :

First stage :
department = aude and animal = yes and babyBed = 1 and area = narbonne

Second stage :
department = aude and animal = yes and babyBed = 1 and area != narbonne

Third stage :
department = aude and animal = yes and babyBed != 1 and area = narbonne

Fourth stage :
department = aude and animal = yes and babyBed != 1 and area != narbonne

7 Evaluation

Generally the evaluation of the efficiency of query result can be done by both recall and precision measurements [7]. The recall is the ratio of the number of relevant documents retrieved by the query to the total number of relevant documents. This measurement is information retrieval oriented that is not our working context. In fact, in our work, the number of results is similar whether we take into account the user profile or not. Indeed, the different stages of a query Q, when combined with a profile P leeds to evaluate formulas of the form (Q and P) and then (Q and ¬ P). Finally, $\mid Q \mid = \mid$ Q and P $\mid + \mid Q \; and \; \neg P \mid$, assuming that |X| is the cardinality of the result set of the expression X. The precision is the ratio of the number of relevant documents retrieved to the total number of documents retrieved (irrelevant and relevant).

In order to simplify the calculation of precision measurement, we have evaluated 700 documents from the collection of PRETI. For visualisation reasons, we present the result of an evaluation process that took place for 26 documents, knowing that this sample is a representative sample. When submitting the initial query shown in the figure 3 (highest left), there are 21 returned documents (see table 1). When considering the results for initial query and if we study the user profile, there is only the document id 19 suits the user profile at 100%, but this

Table 1. Result of initial query without user profile

rank of document	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
document id	1	2	3	6	7	8	9	10	12	13	14	15	16	17	19	21	22	23	24	25	26

Table 2. Result of enriched query

rank of document	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
document id	19	3	17	1	2	6	8	9	10	12	13	14	15	16	23	24	25	26	7	21	22

document take the 15th rank. Conversely, the document id 7 takes the 5th rank although it, doesn't suit the user profile.

On the other hand, when we enriched the query by user profile, the document id 19 for example took the first place, i.e. the result is adapted at the user (see table 2).

In our case, we have defined the precision by the following equation:

$$Precision_{i,j} = \frac{Number\ of\ document\ adapted\ to\ profile\ in\ stage\ j}{first\ i\ document\ returned} \tag{3}$$

This equation (3) has been used to evaluate precision. We show the charts associated to the initial query and enriched query that take into account the stage 1, stage 2, stage 3 and stage 4 (see figure 3) respectively in figures 4, figure 5, figure 6, and figure 7. The initial query is represented by break line curves, that show the precision measurement for i= 5, 10, 15, 20 or 25, where i is the number of first document returned. We evaluate the precision of this initial query by considering, for each retrieved document, its ranking when the query is enriched, more precisely, the stage at which the document is retreived by the enriched query. That leads to obtain the four series of curves shown in figure 4 to 7. For example, the point 5a in figure 5 shows the results of the initial query by evaluating, in the set of the 10 first documents, thoses that corresponds to the stage 2 of the enriched query. Enriched query results are represented by contain line curves, that show the precision measurement for i= 5, 10, 15, 20 or 25, where i is the number of first document returned.

We can notice that the precision of the returned result is better when the query is enriched by user profile. In figure 4 the precision of result returned by the initial query is null in the beginning, but when the query is enriched by user profile, the precision is very high. So, in this case, the result is better adapted to the user.

Normally the result of the initial query includes some document that doesn't correspond exactly to user profile and that may be shown in the first ranks. In order to put these documents at the end of the result, we have proposed to insert them in the final stage of query enrichment (last evaluated query of the complete enriched query). So, we can see on the figure 7 that the precision of the result of enriched query for the last stage is very low at the beginning, which is an expected outcome.

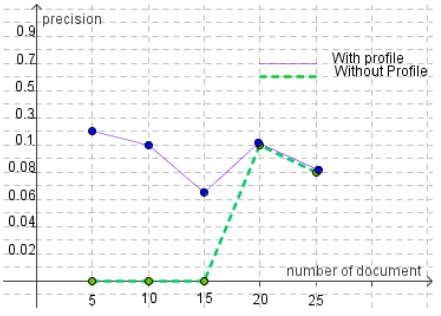


Fig. 4. Precision for the stage 1

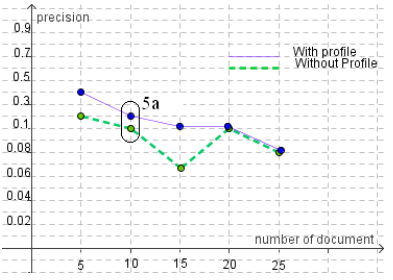


Fig. 5. Precision for the stage 2

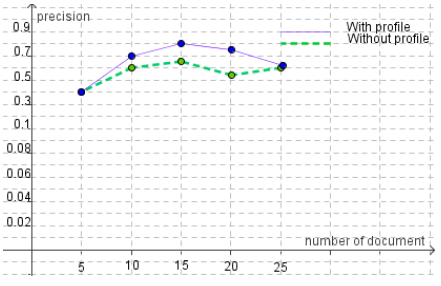


Fig. 6. Precision for the stage 3

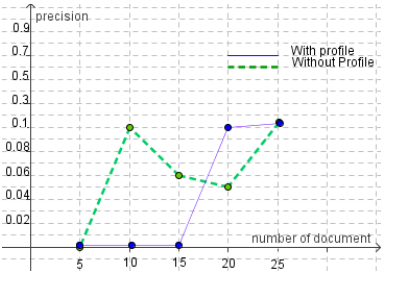


Fig. 7. Precision for the stage 4

8 Conclusion

In this paper we have presented an upstream adaptation process based on an algorithm that consists in enriching the user query on the basis of the user profile in order to adapt the result. We have proposed an evaluation of query enrichment mechanism in the corpus of the PRETI documentary collection. This evaluation compares the precision measurement between the initial query and the enriched query. An issue of our work is to lead to evaluate 2^n queries when "n" user preferences are needed to enrich the query. A first direction of research could be to take into account only a sub set of preferences among n candidates preferences such as for example [13]. Another direction could be to merge all 2^n queries in a single one by changing all conditions expressions on preferences by a single ordering expression.

For the future work, we will improve the algorithm in order to better update the user profile and we will carry out large scale experiments. Moreover, in the enrichment algorithm, we intend to take into account not only the conjunction operator "and", but also the disjunction operator "or" and any combination. On the other hand, we will study the downstream adaptation process, and we intend to define active rules for XML documents which depend on DOM Event mode, in order to adapt the integrality of XML document that consists in documentary units.

References

1. Aiken, A., Hellerstein, J.M., Widom, J.: Static analysis techniques for predicting the behavior of active database rules. ACM Trans. Database Syst. 20(1), 3–41 (1995)
2. Amous, I.: Méthodologies de conception d'applications hypermédia - Extension pour la réingénierie des sites Web. Thése de doctorat, Université Paul Sabatier, Toulouse, France (December 2002)
3. Bra, P.D., Aerts, A., Berden, B., de Lange, B., Rousseau, B., Santic, T., Smits, D., Stash, N.: Aha! the adaptive hypermedia architecture. In: HYPERTEXT 2003: Proceedings of the fourteenth ACM conference on Hypertext and hypermedia, pp. 81–84. ACM Press, New York (2003)
4. Bra, P.D., Houben, G.-J., Wu, H.: Aham: a dexter-based reference model for adaptive hypermedia. In: HYPERTEXT 1999: Proceedings of the tenth ACM Conference on Hypertext and hypermedia: returning to our diverse roots, pp. 147–156. ACM Press, New York (1999)
5. Braga, D., Campi, A.: Xqbe: A graphical environment to query xml data. World Wide Web 8(3), 287–316 (2005)
6. Brusilovsky, P.: Adaptive hypermedia. User Model. User-Adapt. Interact. 11(1-2), 87–110 (2001)
7. Mills, J., Cleverdon, C.W., Keen, E.M.: Factors determining the performance of indexing systems. ASLIB Cranfield Research Project (1966)
8. Conklin, J., Begeman, M.L.: gibis: A hypertext tool for team design deliberation. In: Hypertext, pp. 247–251 (1987)
9. Encelle, B., Jessel, N.: Adapting presentation and interaction with XML documents to user preferences. In: Miesenberger, K., Klaus, J., Zagler, W.L., Burger, D. (eds.) ICCHP 2004. LNCS, vol. 3118, pp. 143–150. Springer, Heidelberg (2004)
10. Halasz, F., Schwartz, M.: The dexter hypertext reference model. Commun. ACM 37(2), 30–39 (1994)
11. Kobsa, A., Koenemann, J., Pohl, W.: Personalised hypermedia presentation techniques for improving online customer relationships. Knowl. Eng. Rev. 16(2), 111–155 (2001)
12. Koch, N., Kraus, A.: The expressive power of uml-based web engineering (2002)
13. Koutrika, G., Ioannidis, Y.: Personalization of queries in database systems. In: ICDE 2004: Proceedings of the 20th International Conference on Data Engineering, p. 597. IEEE Computer Society Press, Washington (2004)
14. van der Weide, T., Bommel, P.v.: GAM: A Generic Model for Adaptive Personalisation. Technical Report ICIS–R06022, Radboud University Nijmegen, Nijmegen, The Netherlands, EU (June 2006)
15. Yi, S., Huang, B., Chan, W.T.: Xml application schema matching using similarity measure and relaxation labeling. Inf. Sci. 169(1-2), 27–46 (2005)
16. Zayani, C., Sedes, F., Peninou, A., Canut, M.-F., Amous, I.: Interrogation de documents semi-structurés: extensions basées sur l'adaptation l'utilisateur. In: INFORSID, Hammamet - Tunisie, 1-03 juin 06, pp. 97–112. Hermés Science Publications (June 2006), `http://www.hermes-science.com/`

Extending SOA with Semantic Mediators

Patrício de Alencar Silva[1], Ulrich Schiel[1], Cláudia M.F.A. Ribeiro[2], and José Eustáquio Rangel de Queiroz[1]

[1] Universidade Federal de Campina Grande
Av. Aprígio Veloso, 882, 58.101-970 Campina Grande – PB – Brazil
{patricio,ulrich,rangel}@dsc.ufcg.edu.br
[2] Universidade do Estado do Rio Grande do Norte
Itapetinga, 1430 Cj. Santarém/Potengi, 59.124-400 – RN – Brazil
{cmfar}@uern.br

Abstract. Mediators are critical for service scenarios, since providers and consumers frequently have opposite goals, e.g. to maximize profit with minimal resource (provider perspective) against to maximize satisfaction with minimal cost (consumer perspective). Nonetheless, semantic mediation is not explicitly taken into account in the Service Oriented Architecture (SOA). In this paper, it is presented an extension of SOA which includes the mediator as first-level entity, besides the client, the provider, and the registry of the services. The assignment of semantics to information that is processed by the mediator through ontologies, and the formalization of mediator's behavior are included among the advantages of the proposal.

Keywords: Semantic Mediators, Semantic Web Services, Service Discovery, Service Composition, Formal Methods.

1 Introduction

In general service oriented applications are potential source of conflict since two opposite parties - clients and providers - are involved. Divergences around the maximization of satisfaction with minimal cost (by the client), against the minimization of resource usage with maximum profit (by the provider) may lead to impasses that require some external help. Usually, mediators play that role bringing together these parties towards an agreement that explicitly considers individual needs but possibly admit reconsiderations. The development of such complex applications commonly requires the adoption of some structured guidelines to be observed.

The Service Oriented Architecture (SOA) [1] and Web services technologies have been largely used by developers of Web applications, whose functionality is divided into self-contained units, the services. This architecture is made up by three entities. Besides the client and the provider, the registry is a fundamental part of this architecture. The basic dynamic tasks of SOA entities include the publication of the service inside the registry by the provider, the following search of specific service by the client, and the consequent bind of both during the invocation phase. Eventually, a service composition can also occur when there is not a single registered service that

E. Damiani et al. (Eds.): SITIS 2006, LNCS 4879, pp. 362–371, 2009.

could fully satisfy the client's needs. An opposite situation can also occur, when many services can fulfill the client's needs. In this case, it is critical to decide what service actually is the best, considering the client's viewpoint.

Situations like service discovery and composition naturally encompass mediation, although SOA does not explicitly treat this aspect. Indeed, semantic unification by the use of mediators is not taken into account at all. Only a few contributions emphasize the need for solving semantic mismatches that inevitably emerge in inter-enterprise integration. Therefore, considering the importance of mediation activity in a service scenario, it is reasonable to put mediators in the first-level of service oriented architectures. It is precisely what this paper deals with.

Besides specific techniques, in order to be efficient, the mediator needs information. It includes information about client and provider requirements, as well as information about the service itself. Although the use of mediators is not new, there is a gap between the formalization of the mediator's behavior and the information description treated by the mediator. The described proposal presents an architecture that extends SOA by the inclusion of the mediator as the fourth entity at the first-level. An important characteristic of the proposed architecture is the use of ontologies that serve as a basis for meaningful information and context-aware activities related to the semantic mediation.

2 Mediators and Semantic Web

Mediators are specialized entities used in conflict situations. Among various aspects, mediation commonly includes a controversy, dispute or difference of viewpoints, or a need for decision-making or problem-solving [2,3,4]. Generally, the process consists of the following stages: (i) inclusion of a mediator between the involved parties; (ii) definition of grounding rules to be applied in the negotiation process;(iii) identification of similarities not explicitly declared between them; (iv) translation of respective subjective expectations into more objective values; (v) identification of options and prediction of the effects resulting from different solutions; (vi) adjustment and refinement of the solutions proposed by the mediator; and (vii) final registration of the agreements into a contract.

In Semantic Web information is given with a well defined meaning through ontologies, which represent a formal and explicit specification of a shared conceptualization about some knowledge domain [5]. However, it is unrealistic to expect a global consensus between people and organizations around a common shared ontology [6].

Semantic heterogeneity represents a typical case in which mediation can be applied. In order to reconcile ontologies, it is necessary to analyze the mismatches between them. Mismatches may occur at the conceptual level, as well as terminological, taxonomical, or syntactical levels [7]. It is necessary to detect and resolve such discrepancies, especially among different semantics. Correspondences can be established among the source ontologies, and overlapping concepts identified.

In SOA semantic mediators can be used to deal with heterogeneity problems that inevitably emerge when interoperability of Web services is attempted. For example, Web services can be described by different ontologies (e.g. WSMO, WSDL-S or OWL-S), different protocols or may have been designed with different goals in mind [8].

Semantic mediators identify implicit similarities, by the use of ontology reconciliation techniques, such as merging, alignment or integration [7]. It is worth noting that mediators must be considered ideally as third-parties, whose the main goal is to approximate different viewpoints, avoiding decisions that could possibly privilege one of the involved parties.

Substantial work have been done on data mediation systems [9,10,11,12]. However, a proper conceptual setting for semantic mediation in SOA is largely missing. Formal methods, based upon elementary mathematics, can be used to produce precise and unambiguous architectural descriptions, in which information is structured and presented at an appropriate level of abstraction. Hence, reasoning about a specification and attempting to construct proofs about its properties help to detect problems at an early stage of systems development. The process of constructing proofs leads to a better understanding about the requirements upon a system, and can assist in identifying any hidden assumptions.

3 An Ontology-Based Architecture for Mediation in SOA

3.1 First-Level Entities

The conceptual elements of the extended SOA architecture proposal are shown in Fig. 1.

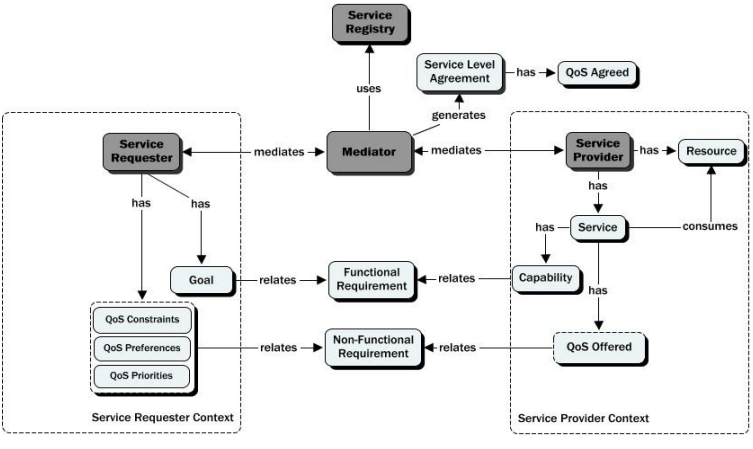


Fig. 1. The Extended Service Oriented Architecture Proposal

The first-level entities, i.e. *Service Requester*, *Service Provider*, *Mediator*, and *Registry* are highlighted. Most of the other concepts are grouped into contexts. In the service requester context, the desired functional requirements are represented by the *goal* concept, while the user's subjective notion of quality is defined by *QoS Preferences* (what the client "wants from" the service), *QoS Constraints* (what "excludes" undesirable services), and *QoS Priorities* (what "differentiates" similar services). On the other hand, in the service provider context, the *capability* concept represents the provided functionality, while *QoS Offered* abstracts a subset of non-functional

requirements related to the service provisioning. It is worth noting that the same service may be offered in different levels of quality, which causes an impact on resource allocation.

In a typical SOA scenario, the service requester directly interacts with the registry, in order to discover potential services that accomplish the desired functionality. In this case, the mediator represents the entity which makes easy this search. This reconciliation process is made on the ontological level. Considering that ontologies are composed basically by classes, properties and axioms, the role of the mediator comprises an automatic verification of conceptual intersections associated with these elements. Additionally, the whole mediation can be considered as a composition of minor mediation processes. The mediation concept in the proposed model abstracts intermediary conflict resolutions, in different levels of complexity.

Mediators make use of well-defined reconciliation laws, valid in some context, in order to approximate different viewpoints and possibly generating a contract that registers the conditions related to the service provisioning. Therefore, any mechanism which makes use of semantics (e.g. semantic web services, intelligent agents or matchmakers) can be used to implement mediators. In order to deal with these complex concepts, a formal specification of the proposed architecture was defined. The benefits of the formalism and specific characteristics of Z notation, used to support formalization in this work, is the subject of the next section.

3.2 Formal Specification

The use of formal methods for describing and verifying properties and behavior of systems is not new [13,14]. The Z notation [15], for example, is a specification language based on the set theory and predicate calculus. Some basic characteristics of Z notation guided its choice as the formalism in this work. The first one is related to its maturity level, recently conveyed into ISO standard [16]. The second one concerns the availability of tools to support formal activities, such as type checkers and theorem provers of formal specifications [17].

The fundamentals of Z notation and types are sets defined at the beginning of the specification. A “given set” is a powerful abstract element of Z, represented by names inside brackets, from which a formal specification begins. Enumerated sets are also permitted in Z notation. In the following, it is enumerated some of the main given sets used to describe the related architectural elements.

$$[Class, Instance, DataType, Parameter, Protocol, Resource, Registry, Preconditions, Postconditions]$$

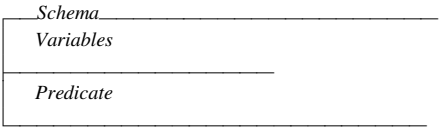
$$Level ::= High \mid Medium \mid Low$$

$$OperationMode ::= Notification \mid OneWay \mid SolicitResponse \mid RequestResponse$$

$$OntologyReconciliationType ::= Alignment \mid Merging \mid Integration$$

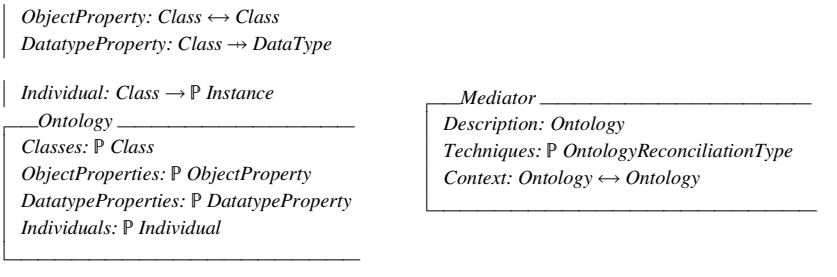
$$ReconciliationResultType ::= ExactMatch \mid PluginMatch \mid SubsumesMatch \mid IntersectionMatch \mid Impasse$$

Other key element of Z specification is the schema. In an analogous way, schema can be considered as a class inside object-oriented paradigm. As the same fashion as class, the schema includes a declarative part to encompass variables and a second part dedicated to the manipulation of variables, the predicate.



Schemas in Z are used to describe both static and dynamic aspects of a system. The static aspects concern the global state of the system and the relationships between its components, namely the invariant relationships. A rigid control over the state integrity is guaranteed by the invariants during any operation execution that changes the global state. Dynamic aspects include all the schema operations that manipulate the elements of the global state [16].

The *Ontology* type is composed by other types including power sets (ℙ) of *Class*, *DatatypeProperties* (a partial function of *Class* in *DataType*), *ObjectProperties* (a relation between two classes) and *Individuals* (a function of *Class* in *Instance*). The *Mediator* is represented as a schema, which is composed by *Description*, typed as an *Ontology*; a power set of *Techniques,* that comprises some ontological reconciliation types; and a *Context*, which represents a relationship between two ontologies, i.e. the conceptualizations of involved partners. For the sake of simplicity and space, other variables and schemas will not be presented here.



4 Investigating Scenarios for Mediation

4.1 Relating Mediators and Service Discovery

In general, service discovery comprises the matchmaking between *goals* (from the service requester context) and service *capabilities* (from the service provider context). The semantics of *goal*, as well as Web service *capabilities* can be represented by a set of concepts described in a *functional requirement ontology* [18]. The semantic mediator verifies similarities between goals and capabilities on the conceptual level. In order to consider goals and capabilities to match on the semantic level, these elements have to be interrelated somehow. Precisely spoken, we expect that some set-theoretic relationships between them exist. The most basic set-theoretic relationships that one might consider are in the Fig. 2.

goal = capability	→	Exact Match
goal ⊂ capability	→	Subsumes Match
capability ⊂ goal	→	Plugin Match
goal ∩ capability ≠ ∅	→	Intersection Match
goal ∩ capability = ∅	→	Non-Match

Fig. 2. Set theoretic relationships about *Goals* and *Capabilities*

These set-theoretic relationships provide the basic means for formalizing an intuitive understanding of a match between goals and Web services in the real-world. For this reason, they are considered to some extent already in the literature [19,20]. The ideal situation can be represented when an *exact match* between requester desires and provider offerings occurs. In this situation the mediator identifies that the offered service functionality coincides perfectly with the service requester goals.

Subsumes match occurs when the advertised capabilities form a superset of relevant objects for the requester, as specified in the goals. In other words, the service might be able to fulfill the desired functionality. However it is possible that the service delivers objects that are irrelevant for the requester. When the service capabilities form a subset of requester goals, then a *plugin match* occurs. In other words, the service in general is not able to provide all the desired functionality, but there is a guarantee that no irrelevant objects will be delivered by the service.

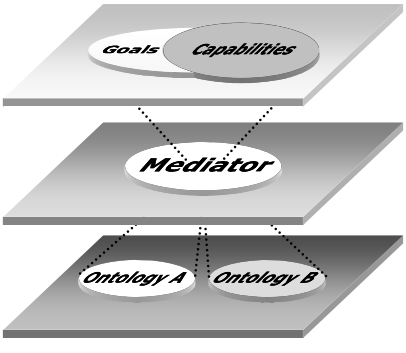


Fig. 3. Semantic mediation in ontology reconciliation: *Intersection Match*

Fig. 3 highlights a case in which the set of service capabilities and the set of requester's goals have an *intersection match*. Thus, the service is not able to deliver all the objects that are relevant for the requester, but at least one such element can be delivered. In this case, a composition of services may be an alternative in order to achieve the desired functionality. Finally, when the service capabilities and the requester goals are disjoint, there is no semantic link. Hence, in the context of service negotiation, it is also considered an *impasse* situation.

As mentioned before, semantic mediation is directly associated with ontology reconciliation techniques. This process involves the alignment of the two basic ontology structures: *classes* and *properties*. The matching cases were formalized by two

operation schemas: *Goal2ServiceClassAlignment* (between classes), and *Goal2ServicePropertyAlignment* (between properties). Since these operations are very similar, only the class alignment is demonstrated. The symbol (Δ) means that this operation changes the global state of the architecture. The predicate section instantiates some local variables, including ontological structures such as *DatatypeProperties* and *ObjectProperties*. The functional requirements are assigned to distinct ontologies followed by a logical concatenation of reconciliation preconditions, as seen below.

Goal2ServiceClassAlignment

$\Delta ARCHITECTURE$

$\exists sr$: ServiceRequester; m: Mediator; sp: ServiceProvider; s: Service; $fr1$, $fr2$: FunctionalRequirement; $o1$, $o2$: Ontology; $c1$, $c2$: Class; $i1$,$i2$: Individual

$| \; sr \in ServiceRequesters \land sp \in ServiceProviders \land m \in Mediators \land s \in sp \,.\, Services$

$\land\, c1 \in o1 \,.\, Classes \land c2 \in o2 \,.\, Classes \land i1 \in o1 \,.\, Individuals \land i2 \in o2 \,.\, Individuals$

$\land\, fr1 = o1 \land fr2 = o2 \land o1 \in sr \,.\, Goal \land o2 \in s \,.\, Capability \land m \,.\, Context = m \,.\, Context \cup \{(o1, o2)\}$

$\Rightarrow (c1, c2) \in EquivalentClass \land (i1, i2) \in SameAs \Rightarrow ReconciliationResult = ExactMatch$

$\lor (c1, c2) \in SubClassOf \Rightarrow ReconciliationResult = PluginMatch$

$\lor (c2, c1) \in SubClassOf \Rightarrow ReconciliationResult = SubsumesMatch$

$\lor (c1, c2) \in ComplementOf \Rightarrow ReconciliationResult = IntersectionMatch$

$\lor (c1, c2) \in DisjointWith \land (i1, i2) \in DifferentFrom \Rightarrow ReconciliationResult = Impasse$

$\bullet\; ServiceRequesters' = ServiceRequesters$

$\land\, ServiceProviders' = ServiceProviders$

$\land\, ServiceLevelAgreements' = ServiceLevelAgreements$

$\land\, Mediators' = Mediators \cup \{m\}$

The schema closes with a state change, with the inclusion of a new local mediator that is created to deal with the service discovery context, according to the *DatatypeProperties* and *ObjectProperties*. Considering these formal statements, the entire discovery phase could be represented by a sequential composition of the two before mentioned operation schemas. The following statement summarizes this aspect.

$$DiscoveryPhase \mathrel{\widehat{=}} Goal2ServiceClassAlignment \fatsemi Goal2ServicePropertyAlignment$$

4.2 Relating Mediators and Service Composition

The partial response to the required functionality justifies the use of composite services. Service composition requires the description of each service, so that other services can understand its features and learn how to interact with [21]. Basically, the service description includes: *domain* information, represented by domain ontologies; a set of *operations,* including aspects related to the message interchange; *bindings,* that defines message formats and protocol details for service invocation. *Capability* describes the business functionalities offered by the service operations. Other elements such as *inputs, outputs, preconditions* and *postconditions* serve as a basis for workflow models. The provider's notion of quality of service is represented by *QoSOffered* (a function of a set of non-functional requirements in a level of quality).

Service composition occurs in different levels, from the binding level to the QoS one [21]. The proposed model aggregates the composition preconditions in two phases: *OperationCompositionPhase* and *ServiceCompositionPhase*. The first one relates the mediation process in the reconciliation of the syntactical features (e.g. mode and message composability). The second one relates semantic features, in which a mediator verifies intersections in binding and domain aspects of the services, according to the reconciliation laws (e.g. *EquivalentClass* and *SameAs* axioms), as seen below.

Message

$Unit: Parameter \rightarrow Ontology$
$Role: Parameter \rightarrow Ontology$
$MessageDatatype: Parameter \rightarrow DataType$

$\text{ran } Unit \cap \text{ran } Role = \emptyset$

Operation

$Description, Domain: Ontology$
$Mode: OperationMode$
$Input, Output: Message$
$Functionality: \mathbb{F}\ FunctionalRequirement$

Service

$Domain: Ontology$
$Operations: \mathbb{P}\ Operation$
$Bindings: \mathbb{F}\ Protocol$
$Input, Output: Parameter$
$Preconditions: \mathbb{P}\ Preconditions$
$Postconditions: \mathbb{P}\ Postconditions$
$Capability: \mathbb{F}\ FunctionalRequirement$
$QoSOffered:$
$NonFunctionalRequirement \nrightarrow Level$

OperationCompositionPhase

$\Delta ARCHITECTURE$

$\exists op1, op2: Operation; m: Mediator; o1, o2: Ontology; c1, c2: Class; i1, i2: Individual$
$\mid m \in Mediators \land (op1, op2) \in isModeComposableWith$
$\land (op1 . Input, op2 . Output) \in isMessageComposableWith$
$\land (op1 . Output, op2 . Input) \in isMessageComposableWith$
$\land o1 = op1 . Domain \land o2 = op2 . Domain \land c1 \in o1 . Classes \land c2 \in o2 . Classes$
$\land i1 \in o1 . Individuals \land i2 \in o2 . Individuals \land m . Context = m . Context \cup \{(o1, o2)\}$
$\Rightarrow (c1, c2) \in EquivalentClass \land (i1, i2) \in SameAs \Rightarrow ReconciliationResult = ExactMatch$
$\lor (c1, c2) \in SubClassOf \lor (c2, c1) \in SubClassOf \Rightarrow ReconciliationResult = PluginMatch$
$\lor (c1, c1) \in ComplementOf \Rightarrow ReconciliationResult = IntersectionMatch$
$\bullet\ ServiceRequesters' = ServiceRequester \land ServiceProviders' = ServiceProvider$
$\land ServiceLevelAgreements' = ServiceLevelAgreement \land Mediators' = Mediator \cup \{m\}$

ServiceCompositionPhase

$\Delta ARCHITECTURE$

$\exists op1, op2: Operation; s1, s2: Service; m: Mediator;$
$o1, o2: Ontology; c1, c2: Class; i1, i2: Individual$
$\mid m \in Mediators$
$\land op1 \in s1 . Operations \land op2 \in s2 . Operations \land (s1, s2) \in isBindingComposableWith$
$\land c1 \in o1 . Classes \land c2 \in o2 . Classes$
$\land i1 \in o1 . Individuals \land i2 \in o2 . Individuals \land o1 = s1 . Domain \land o2 = s2 . Domain$
$\land op1 . Domain = s1 . Domain \land op2 . Domain = s2 . Domain$
$\land m . Context = m . Context \cup \{(o1, o2)\}$
$\Rightarrow (c1, c2) \in EquivalentClass \land (i1, i2) \in SameAs \Rightarrow ReconciliationResult = ExactMatch$
$\lor (c1, c2) \in SubClassOf \Rightarrow ReconciliationResult = PluginMatch$
$\lor (c1, c2) \in ComplementOf \Rightarrow ReconciliationResult = IntersectionMatch$
$\bullet\ ServiceRequesters' = ServiceRequester \land ServiceProviders' = ServiceProvider$
$\land ServiceLevelAgreements' = ServiceLevelAgreement \land Mediators' = Mediator \cup \{m\}$

The entire *CompositionPhase* is defined by a schema that aggregates a sequential composition between *OperationCompositionPhase* and *ServiceCompositionPhase*.

$$CompositionPhase \mathrel{\widehat=} OperationCompositionPhase \fatsemi ServiceCompositionPhase$$

5 Conclusion and Future Work

This paper discusses about the inclusion of the mediators as first class citizens in SOA. The proposed model focuses on conceptual aspects related to the role of the mediator in service discovery and composition. It was investigated the use of ontological reconciliation techniques which serves as a basis for meaningful mediation. Considering that mediation is a complex task which crosscuts all the service-related tasks it is proposed a formal approach in order to specify the semantic mediation unambiguously. The formal specification described in this paper focuses on the description of state-based properties related to the mediation process in ontology reconciliation. The Z notation allows reasoning about those specifications using the proof techniques of mathematical logic. We may also refine that specification, yielding another description that is closer to executable code.

In terms of future work, we intend to investigate and analyze what kind of statements need to be formally checked and proven in order to relate the mediation activity to the other service related tasks, including service selection, negotiation, agreement and monitoring. More specifically, we intend to extend the proposed formal model towards proof obligations in how QoS information can be used by the mediator in order to improve the process of ontological reconciliation. These proof obligations, that basically comprise the formulation of theorems and automatic reasoning and simulation of the proposed architecture properties, will serve as the basis for a most complete framework for mediation in several levels in the context of service negotiation and provisioning.

References

1. Booth, D. (ed.): Web Service Architecture. W3C Working Group Note (February 11, 2004), `http://www.w3.org/TR/ws-arch/`
2. Wiederhold, G.: Mediators in the architecture of future information systems. IEEE Computer, 38–49 (1992)
3. Visser, U.: Intelligent Information Integration for the Semantic Web. Springer, Berlin (2004)
4. Sheth, A., Thacker, S., Patel, S.: Complex relationships and knowledge discovery support in the InfoQuilt system. The VLDB Journal 12(1), 2–27 (2003)
5. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American (2001)
6. Gómez-Perez, A., Fernándes-López, M., Corcho, O.: Ontological Engineering: with example from the areas of Knowledge Management, e-commerce and the Semantic Web. Springer, Heidelberg (2004)
7. Hameed, A., Preece, A., Sleeman, D.: Ontology Reconciliation. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies in Information Systems, pp. 231–250. Springer, Heidelberg (2003)

8. Mrissa, M., Ghedira, C., Benslimane, D., Maamar, Z.: Context and semantic composition of web services. In: Bressan, S., Küng, J., Wagner, R. (eds.) DEXA 2006. LNCS, vol. 4080, pp. 266–275. Springer, Heidelberg (2006)
9. Mocan, A., et al. (eds.): WSMO Mediators, Working Draft (September 16, 2005), http://www.wsmo.org/TR/d29/
10. Garcia-Molina, H., et al.: The TSIMMIS Approach to Mediation: Data Models and Languages. Journal of Intelligent Information Systems, 117–132 (1997)
11. Yan, L., Özsu, M.T., Liu, L.: Accessing Heterogeneous Data through Homogenization and Integration Mediators. In: Proceedings of the 2nd International Conference on Cooperative Information Systems, pp. 130–139 (1997)
12. Tomasic, A., Raschid, L., Valduriez, P.: Scalling Access to Heterogeneous Data Sources with DISCO. IEEE Transactions on Knowledge and Data Engineering 10, 808–823 (1998)
13. Dong, J.S., Sun, J., Wang, H.: Z approach to semantic web. In: George, C.W., Miao, H. (eds.) ICFEM 2002. LNCS, vol. 2495, pp. 156–167. Springer, Heidelberg (2002)
14. Cohen, B.: Justification of formal methods for system specification. Software Engineering Journal, 26–35 (1989)
15. Spivey, J.M.: The Z Notation: A Reference Manual, 2nd edn. Prentice Hall International, Englewood Cliffs (1992)
16. ISO/IEC 13586:2002 – Z formal specification notation – Syntax, type system and semantics, http://www.iso.org
17. Saaltink, M.: The Z/EVES system. In: Till, D., Bowen, J.P., Hinchey, M.G. (eds.) ZUM 1997. LNCS, vol. 1212, pp. 72–85. Springer, Heidelberg (1997)
18. Ribeiro, C.M.F.A., Rosa, N.S., Cunha, P.R.F.: An Ontological Approach for Personalized Services. In: Proceedings of FINA 2006 in conjunction with The IEEE 20th International Conference on Advanced Information Networking and Application (2006)
19. Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.P.: Semantic matching of web services capabilities. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 333–347. Springer, Heidelberg (2002)
20. Stollberg, M., Cimpian, E., Fensel, D.: Mediating Capabilities with Delta-Relations. In: Proceedings of the 1st International Workshop on Mediation in Semantic Web Services, Amsterdam, the Netherlands (2005)
21. Medjahed, B., Bouguettaya, A., Elmagarmid, A.K.: Composing web services on the semantic web. VLDB Journal 12(4), 333–351 (2003)

Author Index